
NEU!
Das Magazin für den Home-Computer-Besitzer
computronic
7/ 1. Jahrgang Juli '84
DM 5,50
SFR 5,50
ÖS 45
Spielprogramme
Anwendungs-programme
für
VC-64
Hardcopy
Space-Figther
Data-Generator
TI-99
Lift-Bear
ASC II-Hilfsprogramm
ZX-Spectrum
Matheprogramm
Bongo-Beecatcher
ZX-81
Tonprogramm
Aldebaran
Apple II
Library
Fight
VC-20
Multigraph
All-Rammer
Atari
Startup
Zeilen-Split
Chopper-Fight
Dragon 32
Laser Attack
mit Forth-Kurs Teil 4
Der ZX 81 gibt Töne von sich!
Jetzt mit Abo-Karte!
Werbe Aktion
Billige Kleinanzeigen
Seite 78

Möchten Sie als Software-Autor für den TRONIC-Verlag tätig werden?

Wir würden uns freuen, in Ihnen einen Ansprechpartner für eine künftige intensive Zusammenarbeit zu finden. Der TRONIC-Verlag vereinbart mit seinen künftigen Software-Autoren pro veröffentlichte Seite (einschl. Programmbeschreibung) ein Honorar von **DM 120,—.**

Dieses Entgelt wird fällig, wenn die Redaktion des Verlages sich für eine Veröffentlichung entscheidet. Die Auszahlung erfolgt also nicht erst nach Veröffentlichung in einer unserer Ausgaben, sondern wesentlich früher.

Der Verlag wird vom Autor berechtigt, seine Manuskripte (Programme) zur Darstellung im Heft heranzuziehen und abzudrucken.

Einzusenden sind:
- Programmbeschreibung und Programmflußplan
- bespielte Cassette oder Diskette
- Listing

mit Copyright.
- Freiumschlag



Haben Sie Interesse? Haben Sie noch Fragen?

Setzen Sie sich telefonisch mit unserer Redaktion

in Verbindung!

TRONIC-VERLAG
DIE REDAKTION

In diesem Heft

Software Listings

Tips & Tricks

Aktuelles

Programmieren in Forth, Teil 4

Tips + Tricks

Paddle am Commodore 64

Viele Besitzer eines Commodore 64 wissen nicht, daß ihr Computer vier analoge Eingänge besitzt. Diese eignen sich für fast alle handelsüblichen Paddles. Paddles bestehen in der Regel nur aus einem Potentiometer sowie einem Feuerknopf. Je nach Stellung des Reglers wird eine entsprechend große Zahl (0 bis 255) vom Computer ermittelt. Dem Basic-Programmierer ist jedoch leider das Auslesen der Paddles versagt. Das folgende Maschinenprogramm hilft nun, diesen Mangel zu beheben:

```
60000 DATA 120,169,128,32,236,207,142,60,3,140
60001 DATA 61,3,173,0,220,41,12,141,159,2,169
60002 DATA 64,32,236,207,142,62,3,140,63,3,173
60003 DATA 1,220,41,12,141,160,2,169,255,141,2
60004 DATA 220,88,96,141,0,220,9,192,141,2,220
60005 DATA 162,0,220,9,192,141,2,220,162,0,202
60006 DATA 208,253,174,25,212,172,26,212,96
60007 FOR I=53182 TO 53247 : READ D : POKE I,D : NEXT I
60008 PRINT PEEK(830) : REM PADDLE 1 AM PORT 1
60009 PRINT PEEK(831) : REM PADDLE 2 AM PORT 1
60010 PRINT PEEK(672) : REM TASTEN VOM PADDLEPAAR A
60011 PRINT PEEK(828) : REM PADDLE 1 AM PORT 2
60012 PRINT PEEK(829) : REM PADDLE 2 AM PORT 2
60013 PRINT PEEK(671) : REM TASTEN VOM PADDLEPAAR B
```

DATA-Generator

In vielen Basic-Programmen werden Maschinenroutinen verwendet, um die nötige Geschwindigkeit zu erreichen. Diese Routinen werden dann oft als „DATA-Zeilen“ in das Progamm eingefügt.

Wer jedoch schon einmal ein längeres Maschinenprogramm programmiert hat, weiß, wie mühselig es ist, das Programm in „DATA-Zeilen“ zu übersetzen.

Das hier vorgestellte Programm nimmt uns die gesamte Arbeit ab. Hat man ein Maschinenprogramm im Speicher stehen, so lädt und startet man den DATA-Generator. Nach der Eingabe der ersten Zeilennummer sowie Anfangs- und Endadresse des Maschinenprogramms erstellt das Programm selbständig die erforderlichen „DATA-Zeilen“ sowie die erforderliche Einlese-Schleife. Bei längeren Maschinenprogrammen kann dies einige Minuten dauern. Nach der Erstellung löscht sich der Data-Generator selbst, so daß nur die Data-Zeilen im Speicher bleiben. Man kann nun das Maschinenprogramm wie ein Basic-Programm laden und speichern. Selbstverständlich können auch Sprites oder Bildschirminhalte u. v. m. in „DATA'S“ umgesetzt werden. Obwohl der Data-Generator nur sehr wenige Bytes belegt, wird er sich schnell zu einem unentbehrlichen Werkzeug verwandeln.

```
S

10 PRINT"■■ *** DATA GENERATOR     F.BRALL ***  ■■":POKE53281
,8
11 INPUT"■ZEILENNR. (>22) ";ZN:INPUT"■ANFANGSADRESSE ";A
12 INPUT"ENDADRESSE      ";E
13 PRINT"■10 FORI="A"TO"E":READ DA:POKEI,DA:NEXTI":PRINT"ZN=
"ZN":E="E":A="A;
14 PRINT":GOTO 15":GOTO20
15 PRINT"■";:IFA>ETHEN21
16 PRINT"■"ZN"DATA ";:FORO=1 TO 17:D=PEEK(A):D$=STR$(D):S$="
"
17 FORI=1 TO 4:E$=MID$(D$,I,1):IF E$<>" "THEN S$=S$+E$
18 NEXTI:PRINTS$",";:A=A+1:IFA>E THEN O=17
19 NEXTO:PRINT"■ ":PRINT"■A="A":ZN="ZN+1":E="E":GOTO 15"
20 POKE 631,19:FORI=632TO640:POKEI,13:NEXT:POKE198,9:END
21 PRINT"■";:FORI=11 TO 17:PRINTI:NEXT:PRINT"GOTO 22":GOTO 2
0
22 PRINT"■";:FORI=18 TO 22:PRINTI:NEXTI::PRINT"LIST":GOTO20
1READY.
```

Runden von Zahlen

Viele Freaks wissen eigentlich nicht, daß Computer nur sehr ungenau rechnen können. Dies macht sich insbesondere bei Zahlen mit Nachkommastellen bemerkbar. Mit dem nachfolgenden Programm kann man Zahlen auf beliebige Stellen runden und somit die Ungenauigkeit etwas ausgleichen.

Das Programm kann als Unterprogramm für fast alle Basic-Computer verwendet werden.

```
10 REM  S = Anzahl der Stellen
20 REM  Z = Zahl
30 Z=INT (Z*(10↑S)+0.5)/(10↑S)
40 RETURN :REM Z enthaelt die gerundete Zahl
```

Tips + Tricks

Video Controller VC 20

Im Gegensatz zu vielen anderen Computern besitzt der VC 20 einen Video-Controller. Dieser Controller übernimmt die gesamte Aufbereitung des Fernsehbildes. Da dieser Controller ab Adresse 36864 ($9000) erreichbar ist, lassen sich kaum vorstellbare Effekte erzielen. Will man beispielsweise den Bildschirm verschieben, so ist dies für den Controller keine Schwierigkeit. Mit dem Befehl POKE 36864,X teilt man ihm die horizontale Position, und mit POKE 36865,X die vertikale Position mit. Mit diesem Trick ist es möglich, das Bild an jede Position des Schirmes zu verschieben. Hier noch eine Aufstellung sehr interessanter Adressen:

```
36864  Horizontale Bildposition (normal 12)
36865  Vertikale Bildposition (normal 38)
36866  Spaltenanzahl des Videobildes (normal 150)
36867  Zeilenanzahl des Videobildes (normal 46)
       Bit 0 waehlt 8*8 oder 16*8 Matrix der Zeichen
       Als sinnvolle Anwendung dieses Registers waere
       die Erzeugung einer feststehenden Zeile,welche sich
       nur durch POKE beschreiben laesst,denkbar.
36869  Bit 4-7 dienen zur Bestimmung der Bildschirmadresse.
       Bit 0-3 bestimmt die Lage des Zeichensatzes.
       Die folgende Tabelle zeigt den Zusammenhang der
       ersten 4 Bit's und der Position des Zeichensatzes:
```

Bit's	Adresse	Bemerkung
0000	32768	ROM Zeichensatz GROSS/GRAFIK
0010	34816	ROM Zeichensatz KLEIN/GROSS
1100	4096	Definierbarer Zeichensatz
1101	5120	" "
1110	6144	" "
1111	7168	" "

TI-99/4a

Der TI-99/4a braucht je nach Länge des Programms ca. 500-1100 Bytes für sich selbst. Wird der Platz zum internen Umdaten zu knapp, meldet er sich mit: „MEMORY VOLL IN“! . . .
Mit dem Ext. Basic-Modul läßt sich das etwas ausgleichen, zudem man vor dem Einlesen des Programms folgendermaßen beginnt:
1. Befehl NEW
2. Befehl CALL FILES (1)
3. Befehl OLD CS1 oder RUN „CS1“ oder
OLD DSK 1. Name oder RUN „DSK 1. Name“.

TI-99/4A

Besitzt man eine Diskettenstation für den TI-99/4A ohne Speichererweiterung, dann läßt sich der CALL FILES (X)-Befehl mit in das Programm einbauen.
Zum Beispiel:
1500 REM Neues Programm anwählen
1510 CALL FILES (1)
1520 RUN „DSK 1. Name“
Je nach Menge der Daten läßt sich eine Menge Speicherplatz zusätzlich gewinnen. Mit Speichererweiterung gilt dieser Befehl als ein Unterprogramm (Programmabsturz).

News

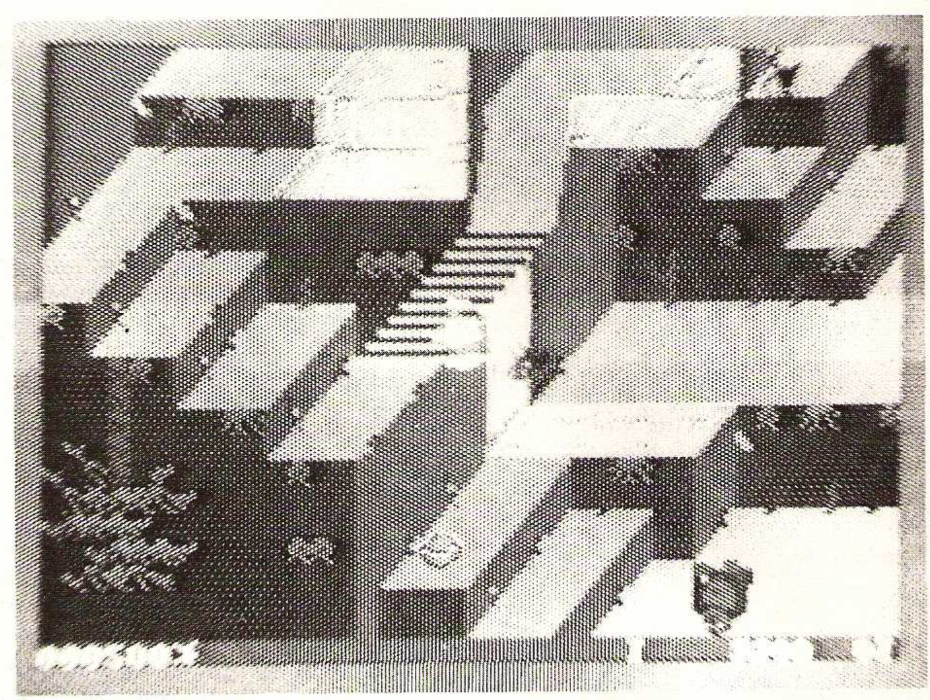
Willkommen in der Zukunft!

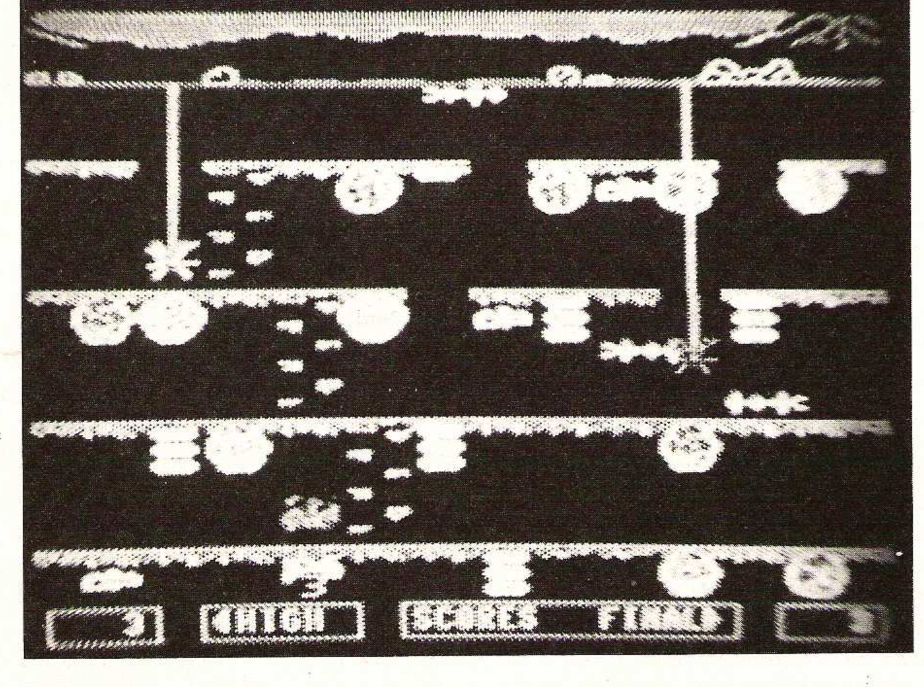
Das Computerzeitalter hat begonnen und wird mehr noch als der Beginn des Industriezeitalters vor rund 200 Jahren die Gesellschaft revolutionieren.
Ob es uns gefällt oder nicht, wir alle werden mit dem Computer leben müssen, werden lernen müssen, mit ihm umzugehen.
Daß auch der Technik aufgeschlossen gegenüberstehende Leute jedoch an „ihrem“ Personal-Computer (Personal = persönlich) scheitern, ist kein Geheimnis. Die Ursache für Angst und Argwohn dem Computer gegenüber liegt grundsätzlich bei nicht ausreichenden Vorkenntnissen, denn auch die inzwischen so zahlreich erschienenen Computer-Bücher lassen den Anfänger in der Regel im Stich.
Hier setzen die HEYNE COMPUTER-BÜCHER an. Die ersten sechs Bände (monatlich wird ab Mai ein Titel erscheinen) führen den Laien systematisch an das unbekannte Wesen Computer heran. Die HEYNE COMPUTER-BÜCHER sind von Fachleuten geschrieben, die sich über den engen Kreis der Freaks und Hacker hinaus verständlich machen können.
Das spätere Programm der HEYNE COMPUTER-BÜCHER wird einzelne Themen für den Fortgeschrittenen vertiefen.
Mit TESTEN SIE IHRE COMPUTER-INTELLIGENZ gibt Dr. Alfred W. Munzert, einer der bekanntesten Testpsychologen Amerikas, auf einfache, verständliche und fast spielerische Weise eine Einführung in Logik und Technik von Computern.
HEYNE COMPUTER-BÜCHER
Dr. Alfred W. Munzert
TESTEN SIE IHRE COMPUTER-INTELLIGENZ
Heyne-Buch Nr. 15/1, DM 9,80
Deutsche Erstausgabe

Software

CONGO BONGO

Der Spieler ist hier der Jäger, der dem Gorilla CONGO BONGO an den Kragen will. Doch der wehrt sich munter seiner Haut, indem er mit Kokosnüssen nur so um sich wirft. Der Jäger muß diesen Wurfgeschossen geschickt ausweichen. Nach Erstürmen des Gipfels verläßt CONGO BONGO den Hügel fluchtartig und rettet sich ans andere Ufer. Der Jäger muß nun versuchen, über den Fluß zu kommen und schwingt dabei nicht wie Tarzan von Liane zu Liane, sondern springt auf dem Rücken der Nilpferde, Wasserpflanzen und Fische ans andere Ufer. CONGO BONGO ist nur für einen Spieler ausgelegt, obwohl die Verpackung etwas anderes aussagt. Wer Frogger kennt, wird feststellen, daß CONGO BONGO keine Überraschungen bietet, obwohl beim ersten Hinsehen die Grafik noch recht interessant ist. Die Figurengestaltung läßt doch einiges zu wünschen übrig. Auch die Farben sind nicht der Natur entsprechend oder haben Sie schon mal einen „rosa" Gorilla gesehen? CONGO BONGO für den Commodore 64 ist bestenfalls ausreichend. rf





Software

DINO EGGS

Beim ersten Einblick in DINO EGGS meint man, ein neues Kletterspiel vor sich zu haben. Man steht am Anfang in einem Tor, in welches man jedesmal zurück muß, wenn die Energie zu Ende ist, oder Sie drei Eier gesammelt haben. Diese werden dann im Tor abgelegt, wobei man selbst mit den Eiern in die Zukunft fliegen kann. Nun erscheinen Sie irgendwo auf der Szene. Um mehr Kraft zu haben, können Sie Blumen essen, wodurch man die Möglichkeit hat, mehr Eier zu tragen.

Beim weiteren Spielablauf werden Sie von Schlangen, Spinnen und allerlei Kleintieren behindert. Diese können Sie mit Steinen beseitigen, welche unüblicherweise an der Decke hängen. Man erhält dadurch Pluspunkte. Sollte man aber die Tiere nicht beseitigen können oder man kommt mit ihnen in Berührung, so verwandelt man sich in eine Spinne und das zweite Leben hat begonnen. Beim Sammeln der Eier sollte man sich aber vor dem DINOSAURIER in acht nehmen, denn dieser wird versuchen, Sie zu zertreten. Weiterhin hat man die Möglichkeit, sich in einen anderen Level zu begeben. Nur dann werden Ihnen Punkte abgezogen, so daß man auch mit Minuspunkten anfangen kann. Ein einfallsreiches Spiel für den Commodore 64, zum Preis von ca. 125 DM im Handel erhältlich. rf

Komm '84

Messe für Bildschirmtext und Microcomputer

In der Zeit vom 17. 5. bis zum 20. 5. 1984 fand in der Gruga-Halle in Essen die Komm '84, eine Ausstellung für den Fachmann und den Computer-Fan statt. Dabei ging es rund um den Mikrocomputer. Im Mittelpunkt dieser Messe stand die Anwendung der Computertechnik für Büro und Beruf sowie für Heim und Hobby.
Wer jedoch gemeint hatte, auf der Messe ein großes Angebot vorzufinden, sah sich sehr enttäuscht. Denn von 63 Anbietern waren ganze 15 Aussteller für den Homecomputer-Bereich. Davon 5 Aussteller für Software und Bücher sowie das Kaufhaus Horten in Essen mit einem reichhaltigen Angebot an Computern und Software. Hier sahen wir auch den Dragon 64 mit Floppy, Drucker und viel Software.



Bei der Firma VIDIS Electronic aus Lingen sahen wir einen neuen Computer. Der „Bit 90", ein Computer »made in Taiwan«, wie geschaffen zum Spielen mit Ton und Graphik. Der „Bit 90" verfügt über Z 80 CPU, erlaubt bis zu 128 Zeichen per Zeile und hat eine hohe Auflösung (256x192 Bildpunkte). Bei 16 Farben und 3 Tongeneratoren mit 5 Oktaven sowie 66 Standardtasten ist der „Bit 90" in zwei Grundversionen zu haben. Mit 18K zum Preis von 554,– DM und in 34K zum Preis von 654,– DM. Mit einem Adapter lassen sich Atari-VCS-Spielkassetten abspielen.
Der Tronic-Verlag wird den „Bit 90" noch testen und in einer unserer Ausgaben ausführlich darüber berichten.
Ein kleines Fazit: Wer mit großer Erwartung auf die Komm '84 ging, wurde ein wenig enttäuscht. Viele Aussteller und Besucher haben mehr Erwartung in diese Messe gestellt. Hier sollten sich Veranstalter und Aussteller einmal mehr einfallen lassen.

Rolf Freitag



Größter Floppy-Disk-Produzent Verbatim mit neuem Management

Mit einem neuen Management für die Region Europa, Afrika und Nahost startet der US-Multi „Verbatim Incorporated" das Fiskaljahr 1984/85. Das kalifornische Unternehmen ist weltweit der größte Hersteller von Floppy-Disks. **Jaime de Barros,** der „Verbatim" in Europa aufbaute, wird abgelöst durch **Antony J. Perry,** 42, welcher als Managing Director Europe, Africa and Middle East die Geschäftstätigkeit des Unternehmens in dieser Region leiten wird. „De Barros" wird in seiner neuen Funktion als Managing Director International Marketing Operations an den Hauptsitz des Unternehmens, nach Sunnyvale, Kalifornien, zurückkehren.

Der neue Verbatim-Generaldirektor für Europa, Afrika und Nahost, Antony J. Perry, kommt vom US-Konzern Union Carbide, wo er als General-Manager der Home Products Division in Genf für den europäischen, den afrikanischen und den Nahost-Raum zuständig war. Zusätzlich war Perry Leiter des Bereichs Planung und Entwicklung am europäischen Hauptsitz von Union Carbide in Genf.

Programmieren in Forth

Forth-Editor

Im letzten Teil haben wir das Definieren von neuen Befehlswörtern kennengelernt. Wir haben auch gelernt, daß in FORTH definieren gleichbedeutend mit Programmieren ist. Um jedoch größere Programme entwikkeln zu können, müssen wir diese abspeichern, laden und editieren können. Dies geschieht in FORTH – im Gegensatz zu Basic – mittels Texteditor. Der Texteditor ist leider in jeder FORTH-Version unterschiedlich zu handhaben. Wir wollen uns deshalb mit unserer Beschreibung auf den weitverbreiteten „FIG-STANDARD“ beziehen.

Im Gegensatz zu vielen anderen Texteditoren wird in FORTH immer nur eine Bildschirmseite editiert. Man spricht dabei von einem „SCREEN“. Diese SCREEN's werden mit einer Nummer versehen und in der Regel auf Diskette gespeichert. Je nach FORTH-Version können 20 bis über 100 solcher Screen's abgespeichert werden. Ein solcher Screen ist in 16 Zeilen zu je 64 Zeichen aufgeteilt. Da bei einigen Computern nur 40 Zeichen darstellbar sind, wird durch ein links/rechts-Scrollen ein größerer Bildschirm simuliert. Ein Beispiel dafür ist „64 FORTH“ von HES. Die meisten FORTH-Versionen besitzen mehrere TEXTPUFFER. Diese erlauben das gleichzeitige Einladen mehrerer „SCREEN's“.

Wollen wir einen Screen auf dem Bildschirm ausgeben lassen, so können wir dies mit dem Befehl: LIST.

Das folgende Beispiel listet Screen Nummer 10:

10 LIST (RETURN)

Ein Screen könnte beispielsweise wie folgt aussehen:

```
0  ( BEISPIEL EINES SCREENES;JUNI 84 )
1
2  : ROUND  SWAP
3           500 + 1000 / +
4           ;
5  : SCALE  392 1000 */MOD ROUND ;
6
7  : M->T   SCALE . CR ;
8
9  : STERN  ." *" ;
10
11 ." DIES IST EIN SCREEN"
12
13
14
15
```

Die Zeilennummern dienen nur zur Orientierung und werden deshalb nicht von allen FORTH-Versionen ausgegeben. Die erste Zeile sollte immer einen Kommentar enthalten, um nicht den Überblick zu verlieren. Der Kommentar beginnt mit einem (Zeichen, gefolgt von einem Leerzeichen und geht bis zum Zeichen). In diesem Kommentar sollte das Datum und die Aufgabe des SCREEN's beschrieben werden. Selbstverständlich können auch in anderen Zeilen noch Kommentare notiert werden.

Wir haben nun gesehen, wie ein beschriebener SCREEN aussehen kann und wollen nun daran gehen, einen SCREEN zu beschreiben.

Zuerst suchen wir uns mit Hilfe eines Handbuches einen freien SCREEN aus. Mit . . . LIST können wir uns überzeugen, daß der SCREEN leer ist. Ist beispielsweise der SCREEN Nummer 10 unbeschrieben, so geben wir folgenden Befehl ein: 10 EDIT (RETURN).

Nun befinden wir uns im eigentlichen Editor. Dieser ist im FIG-STANDARD zeilenorientiert. Das bedeutet, daß der Text mit Hilfe mehrerer Kurzbefehle editiert bzw. erstellt wird.

Die folgende Aufstellung zeigt die Funktion der Standard-Befehle:

LINE n --» adr

Dieser Befehl übergibt dem Stack die Adresse der Zeilennummern. Ist der entsprechende Screen nicht im Puffer, so wird dieser von der DISK eingelesen.

-MOVE adr --» n

Die Textzeile ab Adresse adr wird in Zeile n kopiert.

H n --»

HOLD übernimmt die Textzeile n nach PAD und hält sie dort zur Ausgabe bereit.

E n --»

ERASE löscht Zeile Nummer n und füllt diese mit Leerzeichen (Blanks) aus.

S n --»

Dieser Befehl fügt vor Zeile n eine Leerzeile ein. Nachfolgende Zeilen werden um eine Zeile nach unten ver-

schoben. Die unterste Zeile geht verloren.
D n --»
Die Zeile n wird gelöscht. Dadurch wird der nachfolgende Text um eine Zeile angehoben.
T n --»
Zeile Nummer n wird auf dem Schirm ausgegeben.
L n --»
Der zur Zeit bearbeitete Screen wird ausgelistet.
R n --»
Dieser Befehl ist das Gegenstück zum H-Befehl. Die im PAD abgelegte Zeile wird in Zeile n ausgegeben.
P n --»
Der folgende Text wird in Zeile n übertragen. Der alte Zeileninhalt wird überschrieben.
I n --»
Der PAD wird ab Zeile n ausgegeben. Die nachfolgenden Zeilen werden um eine Zeile nach unten verschoben.
CLEAR n --»
Der Screen n wird mit Leerzeichen (Blanks) vollgeschrieben und somit gelöscht.
COPY n1 n2 --»
Dieser Befehl kopiert den gesamten Screen n1 nach Screen n2.
Die nachfolgend aufgestellten Befehle gehören zwar nicht zum Editor, sind jedoch zum Verwalten der Screen's erforderlich.
LIST n1 --»
Screen n1 wird auf dem Schirm ausgelistet.
FLUSH
Alle geänderten Screen's werden auf Diskette abgespeichert.
EDIT n1 --»
Screen n1 wird eingeladen und kann editiert werden.
LOAD n1 --»
Dies ist der wichtigste Befehl. Screen n1 wird in den Puffer eingeladen und dann Zeile für Zeile compiliert. Wird ein Wort mehrmals definiert, so wird dies mit einer Meldung angezeigt. Der Compilier-Vorgang wird in diesem Falle jedoch nicht unterbrochen. Um jedoch unnötigen Speicherverlust zu vermeiden, sollte man das Wörterbuch vor jedem neuen Compilier-Vorgang zurücksetzen. Wie wir wissen, wird dies mit dem Befehl FORGET . . . gemacht.





Sicher wird sich so mancher schon gefragt haben, wie er längere Programme schreiben soll, wenn doch immer nur ein Screen compiliert wird. Des Rätsels Lösung ist ganz einfach:
Mit Hilfe des --»-Befehls kann man beliebig viele Screen's miteinander verbinden und somit sehr lange Programme schreiben.
Der --»-Befehl wird in diesem Fall an das Ende des letzten Screen's gesetzt. Aufgerufen wird jeweils nur der erste Screen. Je nach Datenträger und Anzahl der Puffer dauert der Compiliervorgang bis zu einigen Minuten.
Einige Forth-Versionen besitzen auch die Möglichkeit, das Wörterbuch direkt abzuspeichern.
Im C64 FORTH von HES heißt der Befehl zum Abspeichern des Wörterbuches: DSAVE.

Das Einladen geschieht in diesem Fall mit DLOAD. Das D steht in beiden Fällen für Dictionary und ist gleichbedeutend mit Wörterbuch. Bei der Benutzung dieser beiden Befehle sollte man bedenken, daß nur der compilierte Code angespeichert wird und somit keine Änderungen mehr möglich sind.

Wir haben nun eine Menge über den FORTH-Texteditor erfahren und können somit auch längere Programme schreiben, ohne diese nach dem Ausschalten zu verlieren.

Wie schon erwähnt, ist es von größter Wichtigkeit, seine FORTH-Programme gut zu dokumentieren, da FORTH-Befehle allein sehr unübersichtlich wirken.

Zur Dokumentation ist übrigens an dieser Stelle zu sagen, daß Kommentare unbestreitbar den Vorzug haben, unverrückbar physikalisch an die kommentierte Definition geknüpft zu sein. Daher wird jede Änderung eines Programms auch mit der Aktualisierung der Kommentierung zwangsläufig verbunden sein. Dennoch sollte man die Kommentare auf der Diskette knapp halten. Genauer:

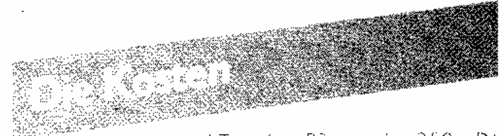
Die mit dem Code verknüpften Kommentare sollten knapp sein, da sonst die Compilierungszeit unnötig erhöht wird.

Daneben ist die Lesbarkeit von Texten innerhalb von Codes relativ schlecht und verschlechtert auch die Lesbarkeit des Codes.

Da die Diskettenlaufwerke von Heimcomputern auch keine unbegrenzte Speicherkapazität besitzen, sollte man sich überlegen, ob man die Dokumentation nicht wie bei Assembler-Programmen getrennt vom Programm vornimmt.

Wie wäre es denn z. B. mit Papier und Bleistift?

Computercamp
Ferienzentrum Schloß Dankern



4 Tage (nur Pfingsten)	350,– DM
1 Woche	550,– DM
2 Wochen	980,– DM
3 Wochen	1480,– DM

Die Preise gelten für Unterbringung **mit Vollpension; die Kurse und fast alle Freizeitmöglichkeiten sind inklusive.**
Sondertarife für Gruppen oder Schulklassen auf Anfrage. Eltern, die ihre Kinder begleiten wollen, sind dazu herzlich eingeladen – rufen Sie uns einfach an!

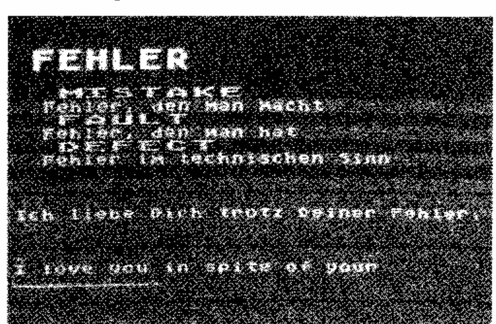
Einsteiger, Fortgeschrittene und „Cracks" werden bei uns von qualifizierten Pädagogen bzw. Informatikern betreut, die es verstehen, **individuell** auf den Wissensstand jedes Teilnehmers einzugehen und Informationen **spielerisch** zu vermitteln.



Die angebotene Palette umfaßt:

- Einführung in Hardware und -Bedienung
- Einführung in die Kommunikation mit dem Computer
- Einführung in die Programmiersprachen
- Die Programmiersprache BASIC von A–Z
- BASIC für Fortgeschrittene
- Perfektionieren von Programmen in BASIC
- Einführung und Programmieren in Maschinensprache

Darüberhinaus aber natürlich **jede Menge detaillierter Information** wie zum Beispiel BASIC-Dialekte, oder wie man einen bereits vorhandenen Computer optimal nutzen kann, etc.
Da der Erfolg der Kurse auch von der Verfügbarkeit der Geräte abhängt, **garantieren wir jedem Teilnehmer ein eigenes Gerät,** das er mindestens **3 Stunden täglich** nutzen kann.
Und – wie gesagt – für individuelle Betreuung ist genauso gesorgt wie für Raum zur Entfaltung von Kreativität und Eigeninitiative – das fördert den wichtigen Erfahrungsaustausch.
Zum Abschluß des Camps erhält jeder Teilnehmer ein **Abschlußzertifikat,** und kann natürlich auch selbst erstellte Programme etc. mit nach Hause nehmen.



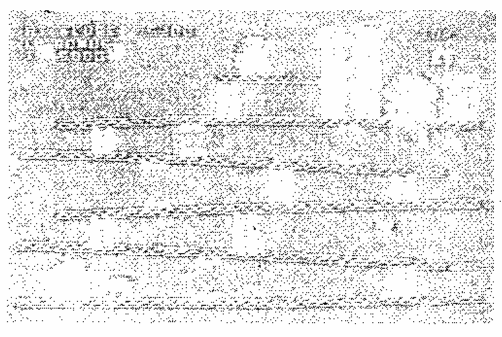
Wir freuen uns schon auf Sie – bis bald!

Fast zuviel, um alles aufzuzählen: **Ponyreiten, Schwimmbad, Minigolf, Tischtennis, Riesenspielplatz** mit **Tarzanschaukel, Westernfort, Fahrradkarussel, Pferderennen, Kletterturm, Riesenrutsche... alles im Preis enthalten!** Darüberhinaus **Kegelbahnen, Autoscooter, Modellflugplatz, Fernsehräume,** und auf dem Wasser **Wasserskilift, Windsurfen, Wasserfahrräder, Tret- und Ruderboote...**

Nicht zu vergessen das 300 Jahre alte **Schloß,** das zur Besichtigung einlädt, genauso wie mehrere **Cafés, Restaurants,** ein **Ferienkino,** eine **Super-Disco...**

Haben wir etwas vergessen? Ach ja – einkaufen kann man auch – sogar sonntags!

Wir finden: Ein insgesamt überzeugendes Angebot, das sicher auch Computerfans begeistern wird.



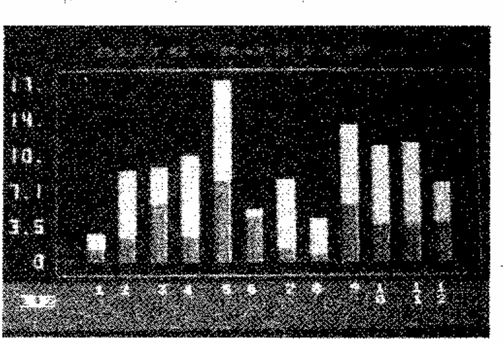
Die Camps beginnen jeweils an einem Samstag. Die Belegung ist möglich für 1 Woche, 14 Tage oder 3 Wochen (Pfingsten nur 4 Tage).

Ostern '84
14.4.–21.4.
21.4.–28.4.

Sommer '84
23.6.–30.6.
30.6.– 7.7.
7.7.–14.7.
14.7.–21.7.
21.7.–28.7.
28.7.– 4.8.
4.8.–11.8.
11.8.–18.8.
18.8.–25.8.
25.8.– 1.9.
1.9.– 8.9.

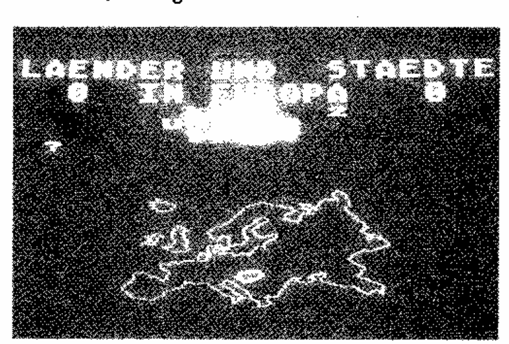
Pfingsten '84
9.6.–12.6.

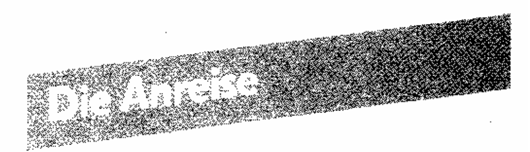
Herbst '84
6.10.–13.10.
13.10.–20.10.
20.10.–27.10.
27.10.– 3.11.



Gewohnt wird nicht in Hotels, Jugendherbergen oder Zeltlagern, sondern in **supergemütlichen Ferienhäusern** (4–6 Personen) aus Holz, die in unmittelbarer Nähe des Schlosses Dankern und des dazugehörigen Dankern-Sees liegen, inmitten eines **150 Hektar großen Ferienzentrums.** Übrigens können Eltern auch gerne ihre Kinder begleiten – Anfrage beim Buchungsbüro genügt.

Frühstück, Mittag- und Abendessen sind inklusive.





Ort der Handlung ist das **Ferienzentrum Schloß Dankern,** 4472 Haren (Ems). Ein riesiges, landschaftlich wunderschön gelegenes Feriengebiet mitten im schönen Emsland. Wie man hinkommt, ist am besten der Karte zu entnehmen. Die Bahn (Bahnhof Haren) ist nur etwa 2 km entfernt. Weitere Fragen beantworten wir gerne – rufen Sie uns bitte an.

Buchungsbüro Computercamp
Ferienzentrum Schloß Dankern
Holztwiete 4 D
2000 Hamburg 52
Tel.: (040) 82 79 42

C64 Hardcopy
Das hier vorgestellte Maschinenprogramm gestattet eine Hardcopy des Commodore-Screens. Hochauflösende Grafik ist mit diesem Programm nicht auszudrucken. Die Bedienung ist denkbar einfach: Nach Eingabe aller 11 Basic-Zeilen startet man das Programm mit RUN. Nun wird es in den entsprechenden Speicherbereich gelegt. Das Programm ist für zwei verschiedene Speicherbereiche ausgelistet. Will man den Bereich ab $9000 benutzen, so sollte man durch Poke 55,0 und Poke 56,9 den Basic-Bereich heruntersetzen. Danach können beliebige Programme eingeladen werden. Durch einen SYS-Aufruf wird die Hardcopy-Routine aufgerufen. Für Profis wurde das Programm auch als Assembler-Listing abgedruckt.
Aufruf des Bereiches $9000 mit SYS 36864
Aufruf des Bereiches $C000 mit SYS 49152



Hardcopybereich $ C 000

```
0 DATA 008,072,234,234,234,234,234,234,234,234,169,000,162,000,160,000
2 DATA 032,189,255,169,004,162,004,160,000,032,186,255,032,192,255,162
3 DATA 004,032,201,255,169,000,160,004,133,251,132,252,162,025,169,013
4 DATA 032,210,255,173,024,208,201,021,240,005,169,017,032,210,255,032
5 DATA 225,255,240,046,160,000,177,251,133,002,041,063,006,002,036,002
6 DATA 016,002,009,128,112,002,009,064,032,210,255,200,192,040,208,230
7 DATA 152,024,101,251,133,251,144,002,230,252,202,208,193,169,013,032
8 DATA 210,255,032,204,255,169,004,032,195,255,104,040,096
12 FORI=49152TOI+124:READX:POKEI,X:S=S+X:NEXT
14 IFS<>16134THENPRINT"FEHLER IN DATAS":END
16 PRINT"HARDCOPY AUFRUF ::SYS 49152 ::"

READY.
```

Hardcopybereich $ 9000

```
1 DATA 008,072,169,143,162,255,133,056,134,055,169,000;162,000,160,000
2 DATA 032,189,255,169,004,162,004,160,000,032,186,255,032,192,255,162
3 DATA 004,032,201,255,169,000,160,004,133,251,132,252,162,025,169,013
4 DATA 032,210,255,173,024,208,201,021,240,005,169,017,032,210,255,032
5 DATA 225,255,240,046,160,000,177,251,133,002,041,063,006,002,036,002
6 DATA 016,002,009,128,112,002,009,064,032,210,255,200,192,040,208,230
7 DATA 152,024,101,251,133,251,144,002,230,252,202,208,193,169,013,032
8 DATA 210,255,032,204,255,169,004,032,195,255,104,040,096
11 FORI=36864TOI+124:READX:POKEI,X:S=S+X:NEXT
13 IFS<>15369THENPRINT"FEHLER IN DATAS":END
15 PRINT"HARDCOPY AUFRUF :: SYS 9*4096 ::"

READY.
```

HARDCOPY

```
  1           ORG $9000      ; STARTADRESSE = 36864
  2NAME       EQU $FFBD      ; SET FILENAMEPARAMETER    :: VC20 ADRESSE - $FFBD
  3FILE       EQU $FFBA      ; SET FILEPARAMETER        :: VC20 ADRESSE - $FFBA
  4OPEN       EQU $FFC0      ; OPEN FILE                :: VC20 ADRESSE - $FFC0
  5CKOUT      EQU $FFC9      ; AUSGABEGERAET SETZEN     :: VC20 ADRESSE - $FFC9
  6CLOSE      EQU $FFC3      ; DATEI SCHLIESSEN         :: VC20 ADRESSE - $FFC3
  7BSOUT      EQU $FFD2      ; AUSGABE EINES ZEICHENS   :: VC20 ADRESSE - $FFD2
  8STOP       EQU $FFE1      ; STOP-TASTE ABFRAGEN      :: VC20 ADRESSE - $FFE1
  9CLRCH      EQU $FFCC      ; EIN/AUSGABE RUECKSETZEN :: VC20 ADRESSE - $FFCC
 10ZSAD       EQU $D018      ; ZEICHENSATZADR.          :: VC20 ADRESSE - $9005
 11BSZW       EQU $02        ; BS - ZWISCHENSPEICHER    :: FREI DEFINIERT
 12BSAD       EQU $FB        ; SPEICHER F. BS -ADRESSE :: FREI DEFINIERT
 13           PHP            ; STATUSREGISTER RETTEN
 14           PHA            ; AKKUINHALT RETTEN
 15           LDA #$8F       ; RAM-BEREICH BEGRENZEN
 16           LDX #$FF       ; ZEILEN 14-17 MIT NOP AUFFUELLEN
 17           STA $38        ; WENN PROGRAMM AB $C000 ABGELEGT WIRD
 18           STX $37        ; (DEZ. 49152) :: ZEILEN 14-17 F. VC20 SELBST FEST-
LEGEN
 19           LDA #00        ; FILENAME PARAMETER SETZEN
 20           LDX #00
 21           LDY #00
 22           JSR NAME
 23           LDA #04        ; DATEI 4 OEFFNEN
 24           LDX #04        ; DRUCKER NENNEN
 25           LDY #00        ; SEKUNDAER ADR. = 00
 26           JSR FILE       ; PARAMETER SETZEN
 27           JSR OPEN       ; DRUCKER OEFFNEN
 28           LDX #04        ; AKTUELLE GERAET SETZEN
 29           JSR CKOUT      ; CMD/AUSGABEGERAET SETZEN
 30           LDA #00        ; LOW-BYTE BS SETZEN
 31           LDY #04        ; HIGH-BYTE BS SETZEN:: VC20 B. 8K =$1E, =>8K =$10
 32           STA BSAD       ; ADRESSE MERKEN
 33           STY BSAD+1
 34           LDX #25        ; ANZAHL D. BS-ZEIHLEN (ZEILENZAEHLER):: VC20 =#23
 35ZAUS       LDA #13        ; NEUE ZEILE ( CR SENDEN)
 36           JSR BSOUT      ; AN DRUCKER AUSGEBEN
 37           LDA ZSAD       ; WELCHER ZEICHENSATZ ?
 38           CMP #21        ; KLEINSCHRIFT EINGESCHALTET ?
 39           BEQ STOP       ; NEIN! AUSGABE FORTSETZEN
 40           LDA #17        ; JA! DRUCKER UMSTELLEN AUF ZEICHENSATZ
 41           JSR BSOUT      ; AN DRUCKER SENDEN
 42STOP       JSR STOP       ; STOPTASTE ABFRAGEN
 43           BEQ END        ; GEDRUECKT? PRG. BEENDEN
 44           LDY #00        ; ZEICHENZAEHLER = 00
 45ENDZ       LDA (BSAD),Y   ; ZEICHEN V. BS HOLEN
 46           STA BSZW       ; U. ZWISCHENSPEICHERN
 47           AND #$3F       ; BIT 6+7 AUSBLENDEN
 48           ASL BSZW       ; 1 * LINKSSCHIEBEN
 49           BIT BSZW       ; BS CODE PRUEFEN
 50           BPL *+2        ; NACH ASCII-CODE WANDELN
 51           ORA #$80       ; BIT 7 SETZEN
 52           BVS *+2
 53           ORA #$40       ; BIT 6 SETZEN
 54           JSR BSOUT      ; UND ZUM DRUCKER SCHICKEN
 55           INY            ; ZEICHENZAEHLER + 1
 56           CPY #40        ; END ZEILE                :: VC20 = #22
 57           BNE ENDZ
 58           TYA            ; < ZEICHENZAEHLER > -> A
 59           CLC            ; JA, ZEIGER AUF NAECHSTE
 60           ADC BSAD       ; ZEILE SETZEN
 61           STA BSAD       ; NEUE BS - ADRESSE SPEICHERN
 62           BCC *+2
 63           INC BSAD+1     ; HIGH-BYTE BS-ADRESSE + 1
```

HARDCOPY

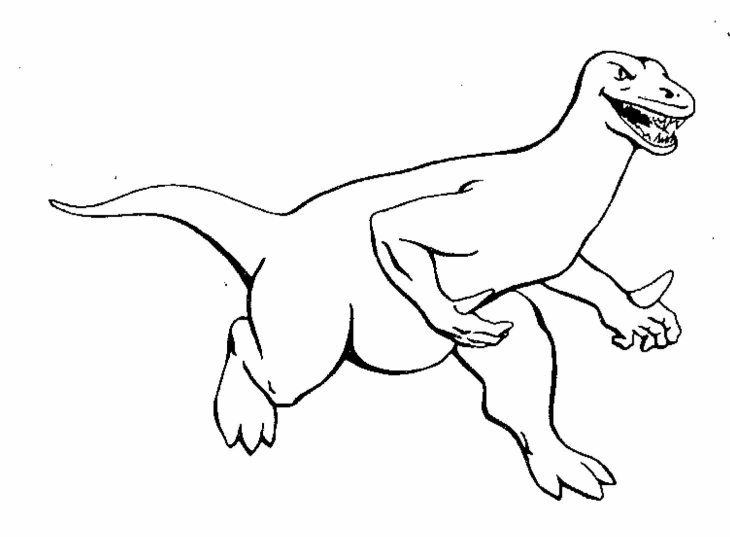
```
64              DEX           ; ALLE ZEILEN AUSGEGEBEN?
65              BNE ZAUS
66              LDA #13       ; CR LADEN UND
67              JSR BSOUT     ; AUSGEBEN
68END           JSR CLRCH     ; AUSGABE AUF BS
69              LDA #04       ; DRUCKERDATEI
70              JSR CLOSE     ; ALLE DATEIN SCHLIESSEN
71              PLA           ; ALTEN AKKUINHALT LADEN
72              PLP           ; ALTEN FLAGREGISTERINHALT (STATUS REGISTER)
73              RTS           ; RUECKSPRUNG
```

VC-64

Space Fighter

Space-Fighter

Dieses Spiel stellt an Sie die Aufgabe, die in drei Wellen angreifenden Monster zu vernichten, wobei mit jeder neuen Welle die Monster schneller werden und sich der Schwierigkeitsgrad erhöht. Trifft ein Monster eine von vier möglichen Basen des Spieles, so wird diese vollkommen zerstört. Die Punkte für jeden Abschuß eines Monsters sind vom jeweiligen Schwierigkeitsgrad abhängig und werden zusammen mit der Zeit als auch der momentan anstehenden Welle im Bildschirm angezeigt.



```
0 GOTO10
1 IFPEEK(V+3)<40THENRETURN
2 TT=PEEK(53269):POKE53269,PEEK(53269)-PEEK(53278):POKE53278
,0
3 IF PEEK(53269)=TTTHENRETURN
4 TR=TR+SG*20:PRINT"SDDDDDDDR"TAB(30);LEFT$(STR$(TR)+"        "
,6):RETURN
5 D=0:K=INT(RND(0)*6):QW=8:IFK<4THENQW=2
6 POKE49154+K,QW:POKE49162+K,EE/2:POKE49170+K,EE/2:AS=1:RETU
RN
7 IFJ<6THENPOKEI,8
8 IFJ>5THENPOKEI,2
9 NEXT:D=0:RETURN
10 REM -------------------------------
11 REM * COPYRIGHT BY REGINALD SCHOLZ *
13 REM -------------------------------
14 GOSUB4000   : REM ERKLAERUNG
15 GOSUB20000  : REM POKE-ROUTINEN
16 TR=0:SS=4:SG=0:WW=0:GOSUB1000
17 TI$="000000"
20 FORSS=1TO4:EE=10-2*SG
25 FORWW=1TO3:ONWWGOSUB1100,1110,1120
```

```
26 PRINT"□":V=53248:POKE53281,14:GOSUB3000
27 PRINT"■"
30 POKEV+21,253
40 POKEV+28,255
50 FORI=0TO7:POKEV+39+I,F(I):NEXT
60 POKEV+37,2:POKEV+38,3
70 POKE2040,11:POKE2041,13
80 FORI=2042TO2047:POKEI,T1:NEXT
100 POKEV,120:POKEV+1,200
110 FORI=2TO7:POKEV+2*I,40+(I-2)*35:POKEV+1+2*I,55:NEXT
111 : REM -------------------
112 : REM **** START HP ****
113 : REM -------------------
115 SYS 49408
120 Q=Q+1:IFQ<20THEN150
130 Q=0:IFPEEK(2042)=T1THENFORI=2042TO2047:POKEI,T2:NEXT:GOT
O150
140 IFPEEK(2042)=T2THENFORI=2042TO2047:POKEI,T1:NEXT
150 IFPEEK(254)=255THENGOSUB2000
160 IF(PEEK(53278)AND2)=2THENGOSUB1
170 IF(PEEK(53278)AND1)=1THENGOTO1800
175 IF(PEEK(53269)AND1)<>1THEN1800
180 IFPEEK(V+21)<4THEN300
190 D=D+1:IFD>100-20*SGTHENGOSUB5
200 PRINT"SQQQQQQQQQQQQQ"TAB(30);TI$"■"
250 GOTO120
300 NEXTWW,SG
390 POKEV+21,0
400 PRINT"QQQQQ    G R A T U L A T I O N !!"
410 PRINT"    ======================"
420 PRINT"QQ■■■■■■ SIE HABEN DAS ZIEL ERREICHT"
430 PRINT"Q UND ALLE INVASOREN VERNICHTET"
440 ZB=1000-VAL(TI$):IFZB<0THENZB=0
450 PRINT"Q DAS ERGIBT EINEN PUNKTESTAND VON :"
460 PRINT"QQ■■■■■■■SCORE     :"TR
470 IFZB>0THENPRINT"Q■■■■■ZEITBONUS:"ZB
480 IF ZB+TR>HITHENHI=ZB+TR:PRINT"QQ   NEUER QHIGHSCORE  : "HI

500 FORI=1TO6000:NEXT
510 GOTO4000
1000 : REM --------------------
1002 : REM **** DEFINITION (ANFANG) **
1004 : REM --------------------
1009 F(0)=6:F(1)=6:F(2)=11:F(3)=6:F(4)=7:F(5)=9:F(6)=1:F(7)=
0
1010 SI=54272:FL=SI:FH=SI+1:TL=SI+2:TH=SI+3:W=SI+4:A=SI+5:H=
SI+6:L=SI+24
1020 RETURN
1100 T1=14:T2=15:GOTO1220
1110 T1=252:T2=253:GOTO1220
1120 T1=254:T2=255:GOTO1220
1220 FORI=49154TO49159:POKEI,1:NEXT
1230 FORI=49162TO49175:POKEI,EE:NEXT
1240 POKE253,4:POKE254,0
1250 POKE49169,1:POKE49161,1:POKE49153,5
1280 RETURN
1797 : REM ----------
1798 : REM ** ZERSTOERUNG **
1799 : REM ----------
1800 POKEL,15:POKEA,47:POKEH,47:POKEW,129
1805 FORI=49152TO49159:POKEI,0:NEXT
1810 FORI=0TO15:POKE53280,I:POKE53281,I:FORK=1TO10
1820 POKEL,15:POKEFH,INT(RND(0)*100):POKEFL,INT(RND(0)*255)
1830 NEXTK,I
1850 POKEW,0:POKEA,0
1853 AA=100+VAL(TI$):AB$=STR$(AA):AA$=RIGHT$(AB$,LEN(AB$)-1)
```

```
1855 TI$=RIGHT$("000000"+AA$,6)
1860 SS=SS-1:IFSS=0THEN2400
1870 WW=WW-1:GOTO300
1997 : REM ----------
1998 : REM ** SCHUSS **
1999 : REM ----------
2000 IF(PEEK(V+21)AND2)=0THENPOKEV+21,PEEK(V+21)+2
2010 POKEV+2,PEEK(V):POKEV+3,PEEK(V+1):POKE49153,5:POKE254,0

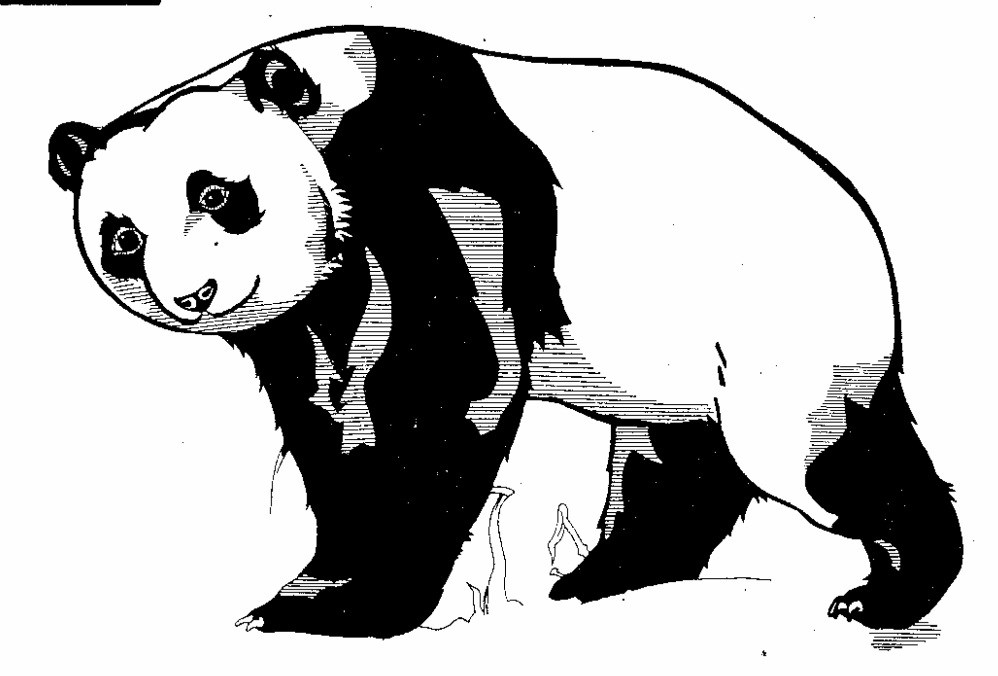
2020 FORI=15TO0STEP-4
2030 POKEL,I:POKEA,15:POKEH,0:POKEFH,40:POKEFL,200:POKEW,129
:NEXT
2040 POKEW,0:POKEA,0
2050 RETURN
2400 : REM ------------------
2402 : REM *** GAME OVER ***
2404 : REM ------------------
2410 PRINT"▣"
2420 FORI=1TO40
2430 PRINT"▣▣▣▣GAME OVER !!!! GAME OVER !!! GAME OVER"
2440 NEXT
2445 POKEV+21,0
2450 PRINT"▣▣▣▣▣▣▣▣         SCORE  : ";TR
2460 IFTR>HITHENHI=TR:PRINT"▣▣         NEUER HIGHSCORE  : "HI
2470 FORI=1TO3000:NEXT
2480 GOTO4000
3000 REM ----------------------
3002 REM ***** AUFBAU *****
3005 REM ----------------------
3007 PRINT"▣"TAB(29)"HIGHSCORE:"
3008 PRINTTAB(30)"▣"HI
3010 PRINT"▣":PRINTTAB(29)"▣▣          ■"
3012 PRINTTAB(29)"▣         ■"
3015 PRINTTAB(29)"▣ PUNKTE:  ■"
3020 PRINTTAB(29)"▣          ■"
3025 PRINTTAB(29)"▣          ■"
3027 PRINTTAB(29)"▣          ■"
3030 PRINTTAB(29)"▣ ZEIT   :  ■"
3040 PRINTTAB(29)"▣          ■"
3045 PRINTTAB(29)"▣          ■"
3050 PRINTTAB(29)"▣          ■"
3060 PRINTTAB(29)"▣ WELLE  :  ■"
3070 PRINTTAB(29)"▣          ■"
3075 PRINTTAB(29)"▣  ■"SG"■ /"WW"■▣  ■"
3080 PRINTTAB(29)"▣          ■"
3090 PRINTTAB(29)"▣";:FORI=1TOSS:PRINT"♠ ";:NEXT
3200 RETURN
4000 : REM ----------------------
4005 : REM **** ERKLAERUNGEN ****
4010 : REM ----------------------
4015 POKE53280,15: POKE53281,15:POKE53269,0
4020 PRINT"▣▣■■■■■■▣     ■ ▣   ■  ▣  ■  ▣ ■  ▣ ■ ▣      ■ ▣
 ■ ▣    ■"
4030 PRINT"■■■■■▣ ■      ▣ ■  ▣ ■  ▣ ■ ▣ ■  ▣ ■   ▣ ■   ▣ ■
  ▣ ■  ▣ ■"
4040 PRINT"■■■■■▣ ■      ▣ ■  ▣ ■    ▣ ■  ▣ ■   ▣ ■   ▣ ■
▣ ■  ▣ ■"
4050 PRINT"■■■■■▣   ■    ▣ ■  ▣ ■ ▣  ■ ▣     ■   ▣ ■   ▣   ■
▣   ■"
4060 PRINT"■■■■■▣ ■      ▣ ■  ▣ ■  ▣ ■ ▣ ■  ▣ ■   ▣ ■   ▣ ■
  ▣ ■ ▣ ■"
4070 PRINT"■■■■■▣ ■     ▣   ■  ▣  ■  ▣ ■  ▣ ■   ▣ ■   ▣    ■
▣ ■  ▣ ■"
4080 PRINT"▣▣▣SPACE":PRINT"▣▣SPACE"
4090 PRINT"▣▣■■■■■■■■■COPYRIGHT BY R. SCHOLZ"
4100 PRINT"▣▣ SCHIESSEN SIE DIE INVASOREN MIT IHRER"
```

```
4110 PRINT"▯ KANONE AB  [ JOYSTICK: PORT 2]"
4120 PRINT"▯ VORSICHT !! DIE EINDRINGLINGE GREIFEN"
4130 PRINT"▯ IN IMMER SCHNELLEREN VERBAENDEN AN."
4140 PRINT"▯■■■■■■■VIEL ERFOLG !!!!"
4150 IFHI<>0THEN4230
4200 PRINT"▯▯▯■■■■■■■■■■■■■■■■■■■■■■BITTE WARTEN"
4220 RETURN
4230 PRINT"▯▯■■■■■'F1' : NEUES SPIEL"
4240 GETA$:IFA$="■"THENGOTO16
4250 GOTO4240
9997 : REM ------------------------
9998 : REM **** SPRITE-DATAS ****
9999 : REM ------------------------
10000 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,8,0,0,8,0,0,8,0,0
,21,0,0,21,0,0
10010 DATA 55,0,32,247,194,32,85,66,20,85,69,28,213,205,31,2
13,253,21,85
10020 DATA 85,22,170,165,21,85,85,3,255,240,59,187,187
10021 REM ------------------
10023 REM SPRITE 13
10026 REM ------------------
10030 DATA 0,8,0,0,8,0,0,55,0,0,29,0,0,53,0,0,29,0,0,34,0,0,
0,0,0,0,0
10040 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0
10050 DATA 0,0,0,0
10051 REM ------------------
10053 REM SPRITE 14
10056 REM ------------------
10060 DATA 1,4,16,4,68,64,0,21,0,0,20,0,10,170,160,40,40,40,
168,40,42,170
10070 DATA 170,170,168,0,42,168,0,42,42,170,168,2,170,128,42
,170,168,32,130
10080 DATA 8,84,130,21,84,130,21,0,130,0,0,130,0,5,65,80,5,6
5,80,0,0,0
10081 REM ------------------
10083 REM SPRITE 15
10086 REM ------------------
10090 DATA 4,17,1,4,17,4,1,17,4,1,17,16,10,170,160,40,40,40,
168,235,42,170,170
10100 DATA 170,168,0,42,168,204,234,42,170,168,2,170,128,42,
170,168,32,130
10110 DATA 8,33,65,72,33,65,72,32,0,8,32,0,8,84,0,21,84,0,21
,0,0,0
10111 REM ------------------
10113 REM SPRITE 252
10116 REM ------------------
10120 DATA0,0,0,2,170,128,42,170,168,163,40,202,163,40,202,1
70,170,170,168
10130 DATA0,42,168,0,42,42,170,168,0,166,0,0,154,0,0,166,0,0
,154,0,0,166,0,0
10140 DATA154,0,2,170,128,2,130,128,2,130,128,3,195,192,3,19
5,192,0,0,0
10141 REM ------------------
10143 REM SPRITE 253
10146 REM ------------------
10150 DATA0,0,0,2,170,128,42,170,168,160,40,10,163,235,202,1
63,235,202,170
10160 DATA170,170,170,170,170,170,170,170,160,0,10,160,221,1
0,160,0,10,170
10170 DATA170,170,42,150,168,0,150,0,2,170,128,2,130,128,2,1
30,128,3,195,192,3
10180 DATA195,192,0,0,0
10181 REM ------------------
10183 REM SPRITE 254
10186 REM ------------------
```

```
10190 DATA1,1,0,4,68,64,0,68,0,0,16,0,0,68,0,10,170,128,35,3
5,32,42,170,160
10200 DATA42,2,160,40,0,160,8,0,128,10,2,128,2,170,0,0,184,0
,0,184,0,0,184
10210 DATA0,2,186,0,170,254,170,170,170,170,68,68,68,0,0,0
10211 REM ------------------
10213 REM SPRITE 255
10216 REM ------------------
10220 DATA0,0,0,0,0,0,0,68,0,0,68,0,0,68,0,10,170,128,35,35,
32,42,170,160
10230 DATA42,170,160,40,0,160,8,0,128,10,170,128,2,170,0,0,1
84,0,0,184,0,0,184
10240 DATA0,2,186,0,170,254,170;170,170,170,17,17,17,0,0,0
10990 : -----------------------
10994 REM MC-ROUTINE
10998 : -----------------------
11000 DATA 169,0,133,251,166,251,189,0,192,133,252,201,0,240
,107,222,16,192
11010 DATA 208,102,189,8,192,157,16,192,216,24,138,101,251,1
70,164,252,185,24
11020 DATA192,41,1,240,15,189,0,208,216,24,101,253,157,0,208
,208,3,76,200,192
11030 DATA185,24,192,41,2,240,15,189,0,208,216,56,229,253,15
7,0,208,208,3,76
11040 DATA200,192,185,24,192,41,4,240,15
11045 DATA 189,1,208,216,56,229,253,157,1,208
11050 DATA208,3,76,200,192,185,24,192,41,8,240,15,189,1,208,
216,24,101,253,157
11060 DATA1,208,208,3,76,200,192,230,251,169,8,197,251,240,3
,76,68,192,76,13
11070 DATA193,162,1,228,251,208,7,169,0,157,0,192,234,234,76

11075 DATA 186,192,234,234,4;208,6,206,0,208,206,0,208,173
11080 DATA0,220,41,16,208,11,169,5,205,1,192,240,4,169,255,1
33,254,76,49,234,0
11085 DATA0,0,0,0,0,0,0
11090 DATA120,169,64,141,20,3,169;192,141,21,3,88,96,173,0,2
20,41,8,208,6
11100 DATA238,0,208,238,0,208,173,0,220,41,4,208,6,206,0,208
,206,0,208,173,0
11110 DATA220,41,16,208,11,169,5,205,1,192,240,4,169,255,133
,254,76
11120 DATA49,234,0,0,0
11130 DATA8,9,1,5,4,6,2,10
20000 : REM --------------------
20004 : REM **** EINPOKEN ****
20006 : REM --------------------
20010 FORI=0TO62:READQ:POKE704+I,Q:NEXT
20020 FORI=0TO62:READQ:POKE832+I,Q:NEXT
20030 FORI=0TO62:READQ:POKE896+I,Q:NEXT
20040 FORI=0TO62:READQ:POKE960+I,Q:NEXT
20050 FORI=0TO62:READQ:POKE252*64+I,Q:NEXT
20060 FORI=0TO62:READQ:POKE253*64+I,Q:NEXT
20070 FORI=0TO62:READQ:POKE254*64+I,Q:NEXT
20080 FORI=0TO62:READQ:POKE255*64+I,Q:NEXT
20100 FORI=49216TO49470:READQ:POKEI,Q:NEXT
20120 FORI=49152TO49176:POKEI,0:NEXT
20130 FORI=49177TO49184:READQ:POKEI,Q:NEXT
21000 RETURN
 READY.
```

Lift Bär



Lift-Bär

Das Programm ist für den TI 99/4 A mit Extended BASIC und Joystick geschrieben.

Nach dem Eintippen, oder laden wird es mit RUN gestartet. Nach einer Pause von ca. 3 Sekunden wird der Bildschirm gelöscht. Dann belegt der Rechner den Grafiksatz um.

Der Bildschirm wird jetzt schwarz, und das Titelbild erscheint in Multicolor Großbuchstaben. Dabei wird Song of Joy gespielt.

Sie können mit der Taste „J“ eine kurze Spielanleitung abrufen. Bei jeder anderen Taste erhalten Sie keine Spielanleitung.

In der Stadt wird ein Neubau errichtet. Es ist kurz nach Feierabend und die Maurer haben ihre Arbeit beendet. Sie haben jedoch einen Teil ihres Werkzeuges vergessen und einen kleinen Bären beauftragt das Werkzeug für sie zu holen. Bei dem Werkzeug handelt es sich um 2 Hämmer, 2 Gabelschlüssel und 2 Schraubendreher.

Die Aufgabe hört sich zwar leicht an, ist aber gar nicht so einfach zu lösen, da erstens der Bär nicht weiß wo das Werkzeug versteckt ist, zweitens die einzelnen Etagen des Neubaues nur mittels auf- und abfahrenden Fahrstühlen zu erreichen sind und drittens nur eine befristete Zeit zur Verfügung steht.

Sie können den Bären mit dem Joystick nach rechts und links bewegen. Achten Sie jedoch darauf, daß die Taste „ALPHA LOCK“ nicht gedrückt ist, da sonst die Joystickabfrage nicht richtig funktioniert. Zum Betreten eines Liftes bewegen Sie den Bären in die Richtung, in der Sie den Lift betreten wollen, immer dann, wenn der Lift gerade vor dem Bären ist und in die gewünschte Richtung fährt.

So lange der Bär im Lift ist, und Sie nicht aussteigen wollen, muß der Joystick in der neutralen Mittelstellung stehen.

Zum Verlassen eines Liftes drücken Sie den Joystick wieder in die gewünschte Richtung.

Der Bär soll ja, wie schon zuvor erwähnt Werkzeug suchen. Das Werkzeug ist in den 12 Fragezeichen versteckt. Zum Aufheben eines Fragezeichens führen Sie den Bären genau darüber und drücken dann den Aktionsknopf am Joystick.

Wenn sich unter dem Fragezeichen ein Werkzeug befindet so ertönt ein Multiklang Gong und das Werkzeug erscheint in der oberen linken Bildschirmecke. Befindet sich unter dem aufgenommenen Fragezeichen jedoch kein Werkzeug, so werden Sie ein Kratzen vernehmen und es kann sein, daß die Wände des Neubaues versetzt werden. Dies kann Ihnen einen günstigeren Weg verschaffen, es kann Ihnen jedoch auch den Weg versperren, so daß Sie einen Umweg zum nächsten Fragezeichen wählen müssen.

Wenn Sie alle 6 Werkzeuge gefunden haben, so bekommen Sie zu Ihren Punkten den verbliebenen Zeitbonus gutgeschrieben, und es geht mit der nächsten Runde weiter.

Insgesamt wird der Schwierigkeitsgrad 7 mal erhöht. Danach bleibt die Schwierigkeit konstant.

Für jeden gefundenen Hammer bekommen Sie 80 Punkte, der Gabelschlüssel bringt 100 Punkte und der Schraubendreher sogar 120 Punkte.

Für alle 5000 erreichten Punkte erhalten Sie ein Bären-Bonusleben gutgeschrieben. Der Bär verliert ein Leben wenn er

1.) zu hoch oder zu tief aus dem Lift springt.
2.) gegen eine vertikale Mauer läuft.
3.) mit dem Lift versucht durch eine Mauer zu fahren.
4.) den Bildschirm nach irgend einer Seite verläßt.
5.) wenn der Zeitbonus auf 0 abgelaufen ist.

Die Fahrstühle fahren immer auf und ab. Bei jedem Richtungswechsel wird der anfänglich 3500 Punkte zählende Zeitbonus um 50 Punkte verringert. Mit steigendem Schwierigkeitsgrad fahren die Lifte immer schneller auf und ab.

Somit läuft auch der Zeitbonus immer schneller, und das Ein- und Aussteigen wird immer schwieriger. Nachfolgend ist das Programm in groben Zügen beschrieben. Den Exakten und strukturierten Programmaufbau können erfahrene Programmierer der beigefügten Zuordnungstafel entnehmen.

```
100 !@P-
110 REM ****************************
120 REM *    L I F T  -  B A E R    *
130 REM *--------------------------*
140 REM * Ein Spiel fuer TI 99/4 A  *
150 REM * in TI Extended BASIC      *
160 REM * mit Joystick zu steuern   *
170 REM *--------------------------*
180 REM * (c) 1984 by Volker Becker *
190 REM *     Steinbacher Str. 10   *
200 REM *     6370 Oberursel 6      *
210 REM ****************************
220 REM
230 REM
240 !@P+
250 OPTION BASE 1
260 DIM VFELD(6),XFELD(9),SF(4),WFELD(3)
270 GOTO 310 :: A :: B :: BER :: BONUS :: BV :: C :: F :: I :: K :: MOT :: P ::
RUNDE :: S :: ST :: SVAR :: T :: V :: VWERT :: X :: X1 :: X2 :: Y :: Z
280 CALL ASCIISET :: CALL BAER :: CALL CLEAR :: CALL COINC :: CALL COLOR :: CALL
 DELSPRITE :: CALL EINLEITUNG :: CALL GCHAR :: CALL GONG :: CALL HCHAR
290 CALL JOYST :: CALL KEY :: CALL LOCATE :: CALL MAGNIFY :: CALL MELODY :: CALL
 MOTION :: CALL PATTERN :: CALL POSITION :: CALL PUNKTE :: CALL SCREEN
300 CALL SONGOFJOY :: CALL SOUND :: CALL SPRITE :: CALL TITEL :: CALL VCHAR :: C
ALL ZEIT :: !@P-
310 CALL CLEAR :: CALL COLOR(5,1,16,6,1,16,7,1,16,8,1,16):: CALL COLOR(3,16,1,4,
16,1,2,14,1,9,6,16)
320 RANDOMIZE
330 CALL CLEAR :: CALL ASCIISET
340 CALL CLEAR :: CALL TITEL(XFELD(),5,SF(),0,2):: CALL CLEAR
350 CALL KEY(5,T,S):: IF T=74 OR T=106 THEN CALL EINLEITUNG
360 CALL CLEAR
370 CALL SCREEN(2)
380 FOR I=4 TO 20 STEP 4
390 CALL HCHAR(I,4,112,12):: CALL HCHAR(I,18,112,12)
400 NEXT I
410 CALL HCHAR(24,1,113,32)
420 CALL VCHAR(1,1,112,24):: CALL VCHAR(1,32,112,24)
430 !@P+
440 DATA 1,8,120,1,12,128,5,6,128,5,11,136,5,23,120,5,27,136,9,4,120,9,29,128
450 !@P-
460 DATA 13,9,128,13,13,136,13,18,120,13,24,136,17,12,120,17,22,128,17,27,136,21
,4,136,21,15,128,21,28,120
470 DATA 8,2,120,8,16,120,8,30,136,12,2,128,12,16,136,12,30,128,16,2,136,16,16,1
28,20,30,120
480 RESTORE 440
490 FOR I=1 TO 18 :: READ A,B,C :: CALL VCHAR(A,B,C,3):: NEXT I
500 FOR I=1 TO 9 :: READ A,B,C :: CALL HCHAR(A,B,C,2):: NEXT I
510 CALL COLOR(11,7,2,13,7,2,12,1,1,14,1,1,9,6,16):: CALL SCREEN(5)
520 RUNDE=1 :: BER=4 :: Z=3500 :: P=0 :: BONUS=1 :: CALL PUNKTE(P):: CALL ZEIT(Z
):: CALL BAER(BER)
530 CALL MAGNIFY(3)
540 CALL SPRITE(#2,116,13,177,9,#3,116,13,1,121,#4,116,13,177,233)
550 MOT=RUNDE+4 :: CALL MOTION(#2,-MOT,0,#3,MOT,0,#4,-MOT,0)
560 DATA 3,10,3,26,7,9,7,25,11,13,11,23,15,11,15,21,19,8,19,25,23,13,23,22
570 GOSUB 820
580 BV=128
590 CALL SPRITE(#1,124,11,169,41)
600 ON ERROR 730
610 CALL ZEIT(Z)
620 REM Hauptschleife
630 V=V+1 :: CALL POSITION(#3,X,Y):: IF X>200 THEN GOSUB 940
640 CALL JOYST(1,X,Y):: IF X=0 THEN 660 ELSE IF X=-4 THEN CALL PATTERN(#1,124)EL
SE CALL PATTERN(#1,132)
650 CALL SOUND(-5,523,0)
```

```
660 CALL MOTION(#1,0,2*X)
670 CALL COINC(ALL,ST):: IF ST THEN 970
680 CALL POSITION(#1,X,Y):: X=(X+7)/8+2 :: Y=(Y)/8+1 :: CALL GCHAR(X,Y,X1)
690 IF X1=BV OR X1=112 OR X1=113 THEN 700 ELSE 730
700 CALL GCHAR(X-2,Y,X1):: CALL GCHAR(X-2,Y+1,X2):: IF X1=BV OR X2=BV THEN 730
710 CALL KEY(1,T,S):: IF T=18 THEN 1080
720 GOTO 620
730 FOR I=1 TO 4 :: CALL MOTION(#I,0,0):: NEXT I
740 CALL PATTERN(#1,140)
750 CALL MELODY
760 BER=BER-1 :: CALL BAER(BER)
770 IF BER<0 THEN CALL DELSPRITE(ALL):: BER=0 :: CALL BAER(BER):: CALL TITEL(XFE
LD(),1,SF(),-1,5):: GOTO 370
780 IF SVAR THEN Z=3500 :: SVAR=0 :: GOTO 1370
790 CALL MOTION(#2,-MOT,0,#3,MOT,0,#4,-MOT,0)
800 SVAR=0
810 GOTO 590
820 RESTORE 560
830 FOR I=1 TO 6
840 VFELD(I)=2
850 NEXT I
860 FOR I=1 TO 12
870 READ A,B
880 F=INT(RND*5+1.4):: IF VFELD(F)=0 THEN 880
890 VFELD(F)=VFELD(F)-1
900 CALL HCHAR(A,B,F+95)
910 NEXT I
920 FOR K=1 TO 3 :: WFELD(K)=0 :: NEXT K
930 RETURN
940 IF V<5 THEN RETURN ELSE V=0 :: MOT=-MOT :: CALL MOTION(#2,-MOT,0,#3,MOT,0,#4
,-MOT,0)
950 Z=Z-50 :: CALL ZEIT(Z):: IF Z<=0 THEN SVAR=-1 :: GOTO 730
960 RETURN
970 FOR I=2 TO 4
980 CALL COINC(#1,#I,16,ST):: IF ST THEN 1010
990 NEXT I
1000 GOTO 720
1010 CALL POSITION(#I,X,Y):: CALL LOCATE(#1,X,Y)
1020 IF I/2-INT(I/2)=0 THEN CALL MOTION(#1,-MOT,0)ELSE CALL MOTION(#1,MOT,0)
1030 CALL POSITION(#1,X,Y):: CALL SOUND(-250,523-X,3):: X=X/8+1 :: Y=Y/8+1 :: CA
LL GCHAR(X,Y,X1):: CALL GCHAR(X,Y+1,X2):: IF X1=BV OR X2=BV THEN 730
1040 CALL JOYST(1,X,Y):: IF X=0 THEN 1030
1050 CALL POSITION(#1,X1,Y)
1060 IF X=-4 THEN CALL LOCATE(#1,X1,Y-18)ELSE CALL LOCATE(#1,X1,Y+18)
1070 GOTO 720
1080 FOR I=1 TO 4 :: CALL MOTION(#I,0,0):: NEXT I
1090 VWERT=0
1100 CALL POSITION(#1,X,Y):: X=(X+7)/8+1 :: Y=(Y)/8+1
1110 CALL GCHAR(X,Y,X1):: CALL GCHAR(X,Y+1,X2)
1120 IF X1<102 AND X1>95 THEN VWERT=X1
1130 IF X2<102 AND X2>95 THEN VWERT=X2
1140 IF VWERT=0 THEN 1170
1150 VWERT=VWERT-95
1160 ON VWERT GOSUB 1190,1210,1220,1230,1240,1250
1170 CALL MOTION(#2,-MOT,0,#3,MOT,0,#4,-MOT,0)
1180 GOTO 720
1190 CALL HCHAR(X,Y,32,2)
1200 CALL COLOR(13,7,2,12,1,1,14,1,1):: CALL SOUND(-500,-7,0):: BV=128 :: RETURN
1210 CALL HCHAR(X,Y,32,2):: CALL COLOR(12,7,2,13,1,1,14,1,1):: CALL SOUND(-500,-
7,0):: BV=120 :: RETURN
1220 CALL HCHAR(X,Y,32,2):: CALL COLOR(14,7,2,13,1,1,12,1,1):: CALL SOUND(-500,-
7,0):: BV=136 :: RETURN
1230 CALL HCHAR(X,Y,32,2):: WFELD(1)=WFELD(1)+1 :: GOSUB 1260 :: RETURN
1240 CALL HCHAR(X,Y,32,2):: WFELD(2)=WFELD(2)+1 :: GOSUB 1260 :: RETURN
1250 CALL HCHAR(X,Y,32,2):: WFELD(3)=WFELD(3)+1 :: GOSUB 1260 :: RETURN
1260 CALL HCHAR(WFELD(VWERT-3),VWERT,VWERT+36)
```

```
1270 P=P+20*VWERT :: CALL PUNKTE(P)
1280 IF BONUS*5000<=P THEN BER=BER+1 :: CALL BAER(BER):: BONUS=BONUS+1
1290 CALL GONG
1300 FOR I=1 TO 3
1310 IF WFELD(I)<2 THEN RETURN
1320 NEXT I
1330 RUNDE=RUNDE+1 :: IF RUNDE>8 THEN RUNDE=8
1340 P=P+Z :: CALL PUNKTE(P):: CALL ZEIT(0):: Z=3500
1350 IF BONUS*5000<=P THEN BER=BER+1 :: CALL BAER(BER):: BONUS=BONUS+1
1360 CALL SONGOFJOY(SF())
1370 CALL DELSPRITE(#1):: CALL HCHAR(1,4,32,3):: CALL HCHAR(2,4,32,3)
1380 CALL COLOR(13,7,2,12,1,1,14,1,1)
1390 GOTO 540
1400 !@P+
1410 SUB SONGOFJOY(B())
1420 GOTO 1430 :: AZ :: I :: J :: K :: M1 :: M2 :: STV :: !@P-
1430 DATA 1,195,246,293,391,2,391,493,1,440,523,1,493,587
1440 DATA 1,184,261,293,369,1,493,587,1,440,523,1,391,493,1,329,440
1450 DATA 1,195,246,293,391,2,246,391,1,261,440,1,293,493
1460 DATA 1,184,261,293,369,2,277,493,2,261,440
1470 DATA 1,195,246,293,391,2,391,493,1,440,523,1,493,587
1480 DATA 1,184,261,293,369,1,493,587,1,440,523,1,391,493,1,329,440
1490 DATA 1,195,246,293,391,2,246,391,1,261,440,1,293,493
1500 DATA 1,184,261,293,369,4,261,440,1,195,246,293,391,4,246,391,0
1510 RESTORE 1430
1520 AZ,STV=0
1530 READ K :: IF K=0 THEN SUBEXIT
1540 READ B(1),B(2),B(3),B(4)
1550 FOR I=1 TO K
1560 FOR J=1 TO 4
1570 STV=STV+1 :: IF STV>AZ THEN READ AZ,M1,M2 :: STV=1
1580 CALL SOUND(400,M1,1,M2,0,B(J),5)
1590 NEXT J :: NEXT I
1600 GOTO 1530
1610 !@P+
1620 SUBEND
1630 SUB TITEL(A(),KY,B(),SVAR,FARBE)
1640 GOTO 1650 :: AZ :: I :: J :: K :: M1 :: M2 :: S :: STV :: T :: V :: SPI$ ::
 !@P-
1650 CALL MAGNIFY(2):: IF SVAR=0 THEN CALL CLEAR :: CALL SCREEN(2)
1660 DATA L,I,F,T,B,A,E,R
1670 DATA G,A,M,E,O,V,E,R
1680 CALL HCHAR(6,13,114,8):: CALL HCHAR(7,13,122,8):: CALL HCHAR(8,13,129,8)::
CALL HCHAR(9,13,137,8)
1690 RESTORE 1660
1700 IF SVAR THEN RESTORE 1670
1710 IF SVAR=0 THEN DISPLAY AT(18,3):"SPIELANLEITUNG ? <J : N>"
1720 J=1
1730 FOR K=6 TO 8 STEP 2
1740 FOR I=13 TO 19 STEP 2
1750 READ SPI$ :: CALL SPRITE(#J,ASC(SPI$),FARBE,(K-1)*8+1,(I-1)*8+1)
1760 J=J+1
1770 NEXT I
1780 NEXT K
1790 DISPLAY AT(24,3)SIZE(23):"@ 1984 BY VOLKER BECKER"
1800 DATA 7,10,11,12,3,13,8,14,5
1810 RESTORE 1800
1820 FOR I=1 TO 9
1830 READ A(I)
1840 NEXT I
1850 CALL COLOR(11,A(1),A(2),12,A(3),A(4),13,A(5),A(6),14,A(7),A(8))
1860 RESTORE 1430
1870 AZ,STV=0
1880 READ K :: IF K=0 THEN 1860
1890 READ B(1),B(2),B(3),B(4)
1900 FOR I=1 TO K
```

```
1910 FOR J=1 TO 4
1920 STV=STV+1 :: IF STV>AZ THEN READ AZ,M1,M2 :: STV=1
1930 CALL SOUND(600,M1,1,M2,0,B(J),5):: CALL KEY(KY,T,S):: IF S<>0 THEN 1980
1940 FOR V=9 TO 2 STEP -1 :: A(V)=A(V-1):: NEXT V :: A(1)=A(9)
1950 CALL COLOR(11,A(1),A(2),12,A(3),A(4),13,A(5),A(6),14,A(7),A(8))
1960 NEXT J :: NEXT I
1970 GOTO 1880
1980 CALL DELSPRITE(ALL):: CALL CLEAR
1990 CALL COLOR(11,1,1,12,1,1,13,1,1,14,1,1)
2000 !@P+
2010 SUBEND
2020 SUB PUNKTE(P)
2030 !@P-
2040 DISPLAY AT(1,23)SIZE(5):USING "#####":P :: CALL HCHAR(1,24,38)
2050 !@P+
2060 SUBEND
2070 SUB ZEIT(Z)
2080 !@P-
2090 DISPLAY AT(1,17)SIZE(4):USING "####":Z :: CALL HCHAR(1,18,35)
2100 !@P+
2110 SUBEND
2120 SUB BAER(BER)
2130 !@P-
2140 DISPLAY AT(1,12)SIZE(2):USING "##":BER :: CALL HCHAR(1,13,36)
2150 !@P+
2160 SUBEND
2170 SUB MELODY
2180 GOTO 2190 :: A :: I :: L :: !@P-
2190 DATA 750,195,250,220,500,233,500,195,1000,277
2200 RESTORE 2190
2210 FOR I=1 TO 5 :: READ L,A :: CALL SOUND(L,A,0):: NEXT I
2220 !@P+
2230 SUBEND
2240 SUB GONG
2250 GOTO 2260 :: I :: !@P-
2260 FOR I=0 TO 9 :: CALL SOUND(-150,329,I):: NEXT I
2270 FOR I=0 TO 9 :: CALL SOUND(-150,277,I,329,I+11):: NEXT I
2280 FOR I=0 TO 9 :: CALL SOUND(-150,220,I,277,I+11,329,I+21):: NEXT I
2290 !@P+
2300 SUBEND
2310 SUB EINLEITUNG
2320 GOTO 2330 :: S :: T :: !@P-
2330 CALL CLEAR :: CALL SCREEN(5)
2340 DISPLAY AT(1,1):"DIE MAURER HABEN VERGESSEN  IHR WERKZEUG EIZUSAMMELN UNDDI
ESE AUFGABE MUSS JETZT    DER LIFT BAER UEBERNEHMEN."
2350 DISPLAY AT(5,1):"DAS WERKZEUG IST IN DEN a   VERSTECKT. JE NACH WERKZEUG GI
BT ES :"
2360 DISPLAY AT(8,1):"( =  80 PUNKTE                 ) = 100 PUNKTE              *
= 120 PUNKTE"
2370 DISPLAY AT(11,1):"JEDES WERKZEUG IST 2 MAL    VORHANDEN. WIRD JEDOCH EIN  a
 AUFGENOMMEN IN WELCHEM    KEIN WERKZEUG IST SO WERDEN"
2380 DISPLAY AT(15,1):"DIE WAENDE DES NEUBAUES     VERSETZT."
2390 DISPLAY AT(17,1):"LIFT BAER VERLIERT EIN LEBENWENN ER IM FALSCHEN MOMENT  A
US DEM LIFT SPRINGT, EINE  MAUER BERUEHRT ODER DER"
2400 DISPLAY AT(21,1):"ZEITBONUS ABGELAUFEN IST.   BONUSLEBEN ALLE 5000 PUNKTE.N
EUES SPIEL MIT AKTIONSKNOPF"
2410 CALL KEY(1,T,S):: IF T=18 THEN SUBEXIT ELSE 2410
2420 !@P+
2430 SUBEND
2440 SUB ASCIISET
2450 GOTO 2460 :: A$ :: I :: CALL CHAR :: !@P-
2460 DATA 003C66666666663C,0018381818181830,003C66060C18307E,003C66061C06663C,00
0C1C2C4C7E0C0C,007E607C0606663C,001C30607C66663C
2470 DATA 007E060C18303030,003C66663C66663C,003C66663E060C38,C3BD665E5E66BDC3,FFC
3999981999999,FF83C9C9C3C9C983,FFC3999F9F9F99C3
```

```
2480 DATA FF83C9C9C9C983,FF819F9F839F9F81,FF819F9F839F9F9F,FFC19F9F919999C3,FF
99999981999999,FFC3E7E7E7E7E7C3,FFF9F9F9F9F999C3
2490 DATA FF9993878F879399,FF9F9F9F9F9F9F81,FFBD9981A5999999,FF9989898191919 9,FF
8199999999999981,FF839999839F9F9F,FFC39999999593C9
2500 DATA FF83999983879399,FFC3999FC3F999C3,FF81E7E7E7E7E7E7,FF999999999999C3,FF
999999C3C3E7E7,FF999999A58199DB,FF99DBC3E7C3DB99
2510 DATA FF99DBC3E7E7E7E7,FF81F9F3E7CF9F81,FEFEFE00EFEFEF,FF18FF18FF18FF18,FFFF
FFFF,0000003C40380478,1F306FD0AFA8ABAB,ABABA8AFA0BF80FF
2520 DATA F80CF60BF515D5D5,D5D515F505FD01FF,FEFEFE00EFEFEF,0000004428101 01,FFFFF
FFF,0182020402020018,000003070D1F0307,0F1F1F0F07040C
2530 DATA 000080C0E0C080C,E0F2F4E4D448C,FEFEFE00EFEFEF,FFFFFFFF,,,00000103070301
03,074F2F272B1203,0000C0E0B0F8C0E,F0F8F8F0E0203
2540 DATA FEFEFE00EFEFEF,FFFFFFFF,,,00000040C1D373F,3F1D08,00000060F2FEF8F8,FAF
EF0680478
2550 RESTORE 2460
2560 FOR I=48 TO 57 :: READ A$ :: CALL CHAR(I,A$):: NEXT I
2570 FOR I=64 TO 90 :: READ A$ :: CALL CHAR(I,A$):: NEXT I
2580 FOR I=112 TO 143 :: READ A$ :: CALL CHAR(I,A$):: NEXT I
2590 FOR I=96 TO 101 :: CALL CHAR(I,"FFC399F9F3E7FFE7"):: NEXT I
2600 CALL CHAR(35,"3C4281899191523C183418BCBEFE2436"):: CALL CHAR(38,"080C0EFFFF
0E0C08")
2610 CALL CHAR(40,"187EFE7E18181818301898F87C0E070340E0703C1E1F0F07")
2620 !@P+
2630 SUBEND
```

TI 99

ASC II - DEF

ASC II – DEF
Definition von Charakteren

Schon lange habe ich mich darüber geärgert, daß beim TI 99/4 A die Belegung der Grafik nach sehr mühsamer Hexadezimaler Belegung erzeugt werden kann. Ich habe mir überlegt, ob es nicht eine andere Möglichkeit der Grafikbelegung gibt.

Diese Möglichkeit habe ich gefunden, indem ich die Grafikgenerierung durch den Rechner selbst machen lasse. Dazu brauche ich nur mit dem Cursor in einer vorgegebenen Maske die einzelnen Kästchen schwarz zu färben, und der TI generiert sich seine Grafik selbst. Danach kann ich die so definierte Grafik auf eine Datei abspeichern und in einem Programm aufrufen. Dies hat noch den Vorteil, daß ich dadurch eine Menge Speicherplatz einspare, wodurch ich im Speicher noch längere Programme unterbringen kann.

Ebenso kann ich den graphisch belegten ASC II-Satz auf dem Drucker ausgeben und sogar eine Hardcopy mit dem vom Benutzer definierten Zeichensatz erstellen.

Aus diesen vorgenannten Betrachtungen entstand nun das nachfolgende sehr leistungsfähige Programmpaket für den TI 99/4 A.

Bei diesem Programmpaket wurden die Fähigkeiten des TI voll ausgenutzt und das mit Extended BASIC. Denn bei einer geschickten Programmierung kann auch in EX-BASIC eine hohe Rechengeschwindigkeit erreicht werden.

So werden Anwender eines Diskettenlaufwerkes eine Hardcopy mit vorher definierter ASCII-Datei in ca. 2 1/2 Minuten erstellen können.

Das Programmpaket ist so gestaltet, daß es sowohl mit Diskettenlaufwerk als auch mit Cassettenrecorder gefahren werden kann. Jedoch ist die Verarbeitung mittels Cassette wesentlich langsamer als mit Diskette.

Das Paket wurde extra in mehrere Programme unterteilt, damit keine Speichererweiterung erforderlich ist. Das Programmpaket kann deshalb auch mit der Grundkonfiguration (Konsole, Fernseher und Recorder) gefahren werden.

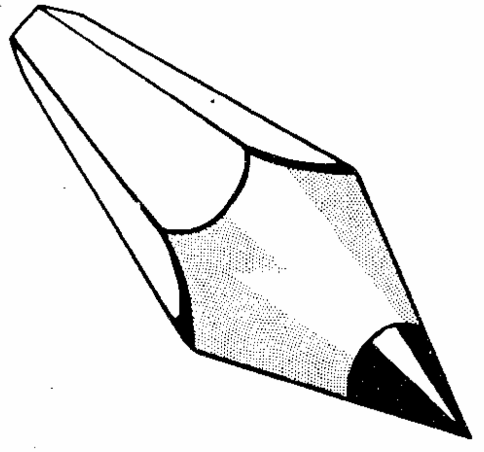
Selbst ohne externen Speicher ist das Programm ASCIIDEF durch seine vielseitigen Grafikmöglichkeiten noch interessant.

Als Drucker wurde ein Seikosha GP 700 A verwendet. Es kann jedoch auch jeder andere Drucker mit 8 DOT Matrix verwendet werden. Dazu sind evtl. nur die Druckeroptionen zu ändern, die aus diesem Grunde als Stringvariable am Anfang der Pro-

gramme ASCIIPRINT und COPYPRINT stehen.
Auf eine Farbsteuerung des Druckers wurde bewußt verzichtet, da dies evtl. Schwierigkeiten bei der Umsetzung auf andere Drucker geben könnte.
Das Programmpaket besteht aus folgenden Programmen:
ASCIIDEF (Definition von Charakteren)
ASCIIPAT (Unterprogramm zur autom. Definition)
ASCIISET (Umbelegung des Grafiksatzes mit Datei)
ASCIIUMW (Umwandeln für die Druckerausgabe)
ASCIIPRINT (Ausdruck des graphischen ASCII Satzes)
COPY (Unterprogramm zur Speicherung d. Hardcopy)
COPYPRINT (Umwandeln u. Ausdruck der Hardcopy)

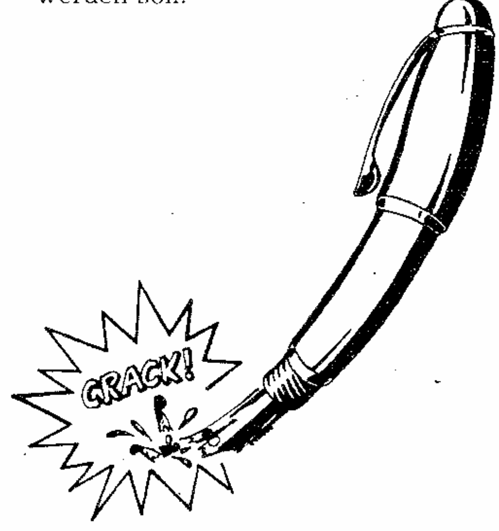
Programm ASCIIDEF
Das Programm ASCIIDEF erlaubt es dem Programmierer, einen eigenen Zeichensatz im Bildschirmdialog zu erstellen sowie einen bereits erstellten Zeichensatz zu verändern.
Die definierten Zeichen können gespiegelt werden sowie nach rechts oder links gedreht werden. Es können Zeichen in einer 8x8-Matrix (1 Charakter) als auch in einer 16x16-Matrix (4 Charaktere für Sprites im MAGIFY(3)-Modus) definiert und editiert werden. Auch kann man einzelne Teile einer 16x16-Matrix verändern.
Die so erstellte Datei kann dann mittels des Unterprogramms ASCIISET zur Umbelegung des Zeichensatzes verwendet werden.
Laden Sie das Programm ein und starten Sie es mit RUN. Das Programm führt Sie in Dialog weiter. Zuerst erscheint auf dem Bildschirm das Wort INITIALISIEREN . . . . , dabei wird die Datei mit dem Standardzeichensatz des TI 99/4 A belegt. Danach erscheint GRAPHIK BELEGEN . . . , dabei wird die im Programm benötigte Grafik generiert. Jetzt erscheint die Frage NEUES CODEFELD ERSTELLEN? Wird die Frage mit ‚N' beantwortet, so kann jetzt ein bereits vorhandenes Codefeld von Diskette oder Cassette geladen werden. Dazu wird nach Aufforderung der Dateiname (z. B.: DSK1, FELD oder CS1) eingegeben.

Ist die Antwort jedoch ‚J', so wird als Codefeld der Standard-Zeichensatz des TI vorausgesetzt. Jetzt wird gefragt ‚16x16-PUNKTMATRIX'. Antworten Sie mit ‚J', so erscheint diese Matrix auf dem Bildschirm. Wenn Sie mit ‚N' antworten, so wird eine 8x8-Matrix ausgegeben.



Als nächstes werden Sie gefragt, ob Sie einen Charakter ändern wollen. Wenn die Antwort ‚J' lautet, so müssen Sie die Nummer des zu ändernden Zeichens angeben. Bei einer 16x16-Matrix kann diese Nummer nur dem ersten oder fünften Zeichen eines der 14 Charaktersätze des Rechners entsprechen, da sonst bei einem späteren Spriteaufruf in anderen Programmen das Bild verdreht wiedergegeben würde. Dies wird jedoch vom Programm kontrolliert und gegebenenfalls mit einer Fehlermeldung quittiert, die dazu führt, daß man sich nochmals dazu entscheidet, ob der Charakter geändert werden soll oder nicht. Der Charakter wird jetzt auf der großen Matrix generiert und das Programm springt in den EDIT-Modus. Wenn keine Änderung eines Charakters erwünscht ist, so springt das Programm bei einer weißen Matrix in den EDIT-Modus.
Im Edit-Modus kann der Charakter definiert werden:
Taste ‚1' färbt das Kästchen schwarz.
Taste ‚0' färbt das Kästchen weiß.
Taste ‚D' bewegt den Cursor nach rechts.
Taste ‚S' bewegt den Cursor nach links.
Taste ‚E' bewegt den Cursor nach oben.
Taste ‚X' bewegt den Cursor nach unten.

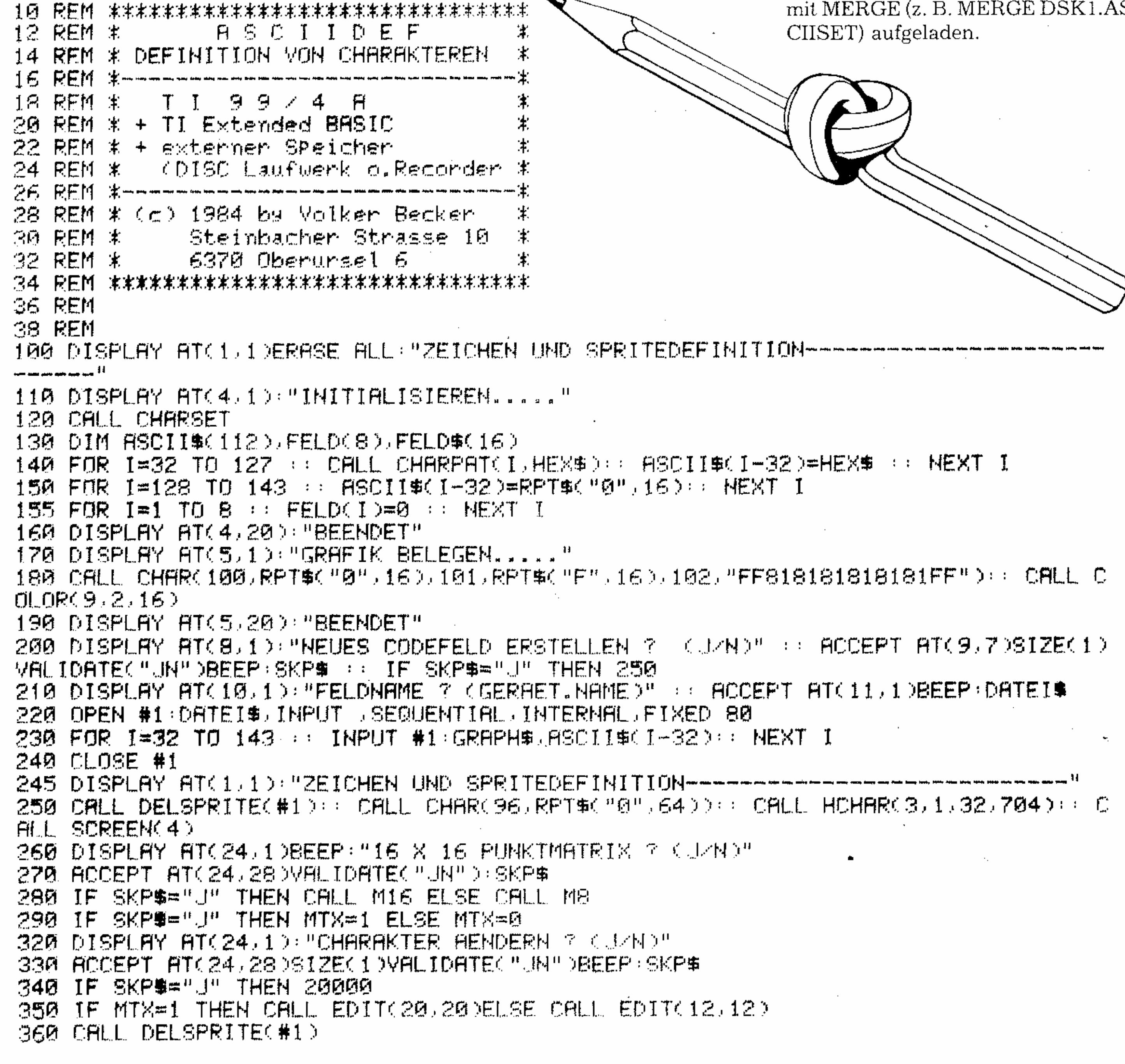
Um den Editiermodus zu beenden, drücken Sie die Tasten ‚FCTN' und ‚BACK' gleichzeitig und Sie erhalten das Menue:
‚1' – ZEICHEN GENERIEREN
‚2' – ZEICHEN SPIEGELN
‚3' – ZEICHENDREHUNG RECHTS
‚4' – ZEICHENDREHUNG LINKS
ZEICHEN UND SPRITEDEFINITION
Bei Auswahl ‚2' wird das generierte Zeichen gespiegelt.
ZEICHEN UND SPRITEDEFINITION
Die Auswahl ‚3' hat zur Folge, daß das Zeichen im Uhrzeigersinn um 90 Grad gedreht wird.
ZEICHEN UND SPRITEDEFINITION
Auswahl ‚4' dreht das Zeichen um 90 Grad gegen den Uhrzeigersinn.
ZEICHEN UND SPRITEDEFINITION
Bei Auswahl ‚1' wird der Hexadezimale Code des definierten Zeichens erzeugt und ausgegeben. Um fortzufahren, müssen Sie die Tasten ‚FCTN' und PROC/D gleichzeitig drücken.
Es erscheint nun die Frage: CHARAKTER ABSPEICHERN. Bei Antwort ‚J' werden Sie aufgefordert, die Kennzahl des Zeichens einzugeben, unter welcher dieses gespeichert werden soll.



Nach sämtlichen 4 Auswahlmöglichkeiten werden Sie jetzt gefragt: WEITERE OPERATIONEN. Diese Frage bezieht sich auf das z. Z. auf dem Bildschirm sichtbare Zeichen.
Wenn Sie die Frage mit ‚J' beantworten, kommen Sie wieder in den EDIT-Modus.

Wird die Frage mit ‚N' beantwortet, so werden Sie nach weiteren Definitionen gefragt. Antworten Sie mit ‚J', um zu der Frage 16x16 PUNKTMATRIX zu kommen. Wenn Sie jedoch mit ‚N' antworten, so haben Sie jetzt die Auswahl, die soeben definierte Datei zu speichern. Auch ohne Speicherung und Weiterverarbeitung der Datei ist das Programm ASCIIDEF ein sehr hilfreiches Werkzeug, welches die mühsame Arbeit der Charakterdefinition in Hexadezimalcode abnimmt.

Wenn Sie bereits ein Programm haben, worin Charaktere belegt sind, so können Sie sich das Codefeld dieses Programms mittels des Unterprogramms ASCIIPAT abspeichern und evtl. auch editieren, ausdrucken oder Hardcopys dieses Programms ziehen. Tippen Sie das Unterprogramm ASCIIPAT zusätzlich in Ihr Programm ein oder laden dieses mit MERGE (nur mit Diskettenlaufwerk) auf.
Um das Codefeld zu erhalten, rufen Sie das Unterprogramm nach der letzten CHAR-Belegung mit CALL ASCIIPAT(„NAME") auf, wobei „NAME" für die entsprechende Datei (z. B. DSK1.FELD" oder „CS1") steht.
Die so erstellte Datei hat genau die gleiche Form wie das Codefeld, welches mittels „ASCIIDEF" erstellt wurde und ist somit vollends kompartibel zu dieser Datei.
Um in einem Programm den Zeichensatz mittels eines zuvor erstellten Codefeldes umzubelegen, wird das Unterprogramm ASCIISET verwendet. Das Programm wird zusätzlich zum normalen Programm eingetippt oder mit MERGE (z. B. MERGE DSK1.ASCIISET) aufgeladen.



```
10 REM *****************************
12 REM *         A S C I I D E F      *
14 REM * DEFINITION VON CHARAKTEREN   *
16 REM *-----------------------------*
18 REM *   T I  9 9 / 4  A            *
20 REM * + TI Extended BASIC          *
22 REM * + externer Speicher          *
24 REM *   (DISC Laufwerk o.Recorder *
26 REM *-----------------------------*
28 REM * (c) 1984 by Volker Becker    *
30 REM *      Steinbacher Strasse 10  *
32 REM *      6370 Oberursel 6        *
34 REM *****************************
36 REM
38 REM
100 DISPLAY AT(1,1)ERASE ALL:"ZEICHEN UND SPRITEDEFINITION--------------------
------"
110 DISPLAY AT(4,1):"INITIALISIEREN....."
120 CALL CHARSET
130 DIM ASCII$(112),FELD(8),FELD$(16)
140 FOR I=32 TO 127 :: CALL CHARPAT(I,HEX$):: ASCII$(I-32)=HEX$ :: NEXT I
150 FOR I=128 TO 143 :: ASCII$(I-32)=RPT$("0",16):: NEXT I
155 FOR I=1 TO 8 :: FELD(I)=0 :: NEXT I
160 DISPLAY AT(4,20):"BEENDET"
170 DISPLAY AT(5,1):"GRAFIK BELEGEN....."
180 CALL CHAR(100,RPT$("0",16),101,RPT$("F",16),102,"FF818181818181FF"):: CALL C
OLOR(9,2,16)
190 DISPLAY AT(5,20):"BEENDET"
200 DISPLAY AT(8,1):"NEUES CODEFELD ERSTELLEN ?  (J/N)" :: ACCEPT AT(9,7)SIZE(1)
VALIDATE("JN")BEEP:SKP$ :: IF SKP$="J" THEN 250
210 DISPLAY AT(10,1):"FELDNAME ? (GERAET.NAME)" :: ACCEPT AT(11,1)BEEP:DATEI$
220 OPEN #1:DATEI$,INPUT ,SEQUENTIAL,INTERNAL,FIXED 80
230 FOR I=32 TO 143 :: INPUT #1:GRAPH$,ASCII$(I-32):: NEXT I
240 CLOSE #1
245 DISPLAY AT(1,1):"ZEICHEN UND SPRITEDEFINITION--------------------"
250 CALL DELSPRITE(#1):: CALL CHAR(96,RPT$("0",64)):: CALL HCHAR(3,1,32,704):: C
ALL SCREEN(4)
260 DISPLAY AT(24,1)BEEP:"16 X 16 PUNKTMATRIX ? (J/N)"
270 ACCEPT AT(24,28)VALIDATE("JN"):SKP$
280 IF SKP$="J" THEN CALL M16 ELSE CALL M8
290 IF SKP$="J" THEN MTX=1 ELSE MTX=0
320 DISPLAY AT(24,1):"CHARAKTER AENDERN ? (J/N)"
330 ACCEPT AT(24,28)SIZE(1)VALIDATE("JN")BEEP:SKP$
340 IF SKP$="J" THEN 20000
350 IF MTX=1 THEN CALL EDIT(20,20)ELSE CALL EDIT(12,12)
360 CALL DELSPRITE(#1)
```

```
370 DISPLAY AT(21,1):"'1' - ZEICHEN GENERIEREN     '2' - ZEICHEN SPIEGELN        '3
' - ZEICHENDREHUNG RECHTS '4' - ZEICHENDREHUNG LINKS"
380 ACCEPT AT(19,26)VALIDATE("1234")SIZE(1)BEEP:SKIP
390 ON SKIP GOSUB 2000,3000,4000,4000
400 CALL HCHAR(21,1,32,128):: CALL HCHAR(19,28,32):: DISPLAY AT(24,1):"WEITERE O
PERATIONEN ? (J/N)" :: ACCEPT AT(24,28)SIZE(1)VALIDATE("JN")BEEP:SKP$
410 IF SKP$="J" THEN 350
420 DISPLAY AT(24,1):"WEITERE DEFINITIONEN (J/N)" :: ACCEPT AT(24,28)SIZE(1)VALI
DATE("JN")BEEP:SKP$ :: IF SKP$="J" THEN 250
430 CALL DELSPRITE(#1):: CALL HCHAR(3,1,32,704):: DISPLAY AT(4,1):"CODEFELD ABSP
EICHERN? (J/N)" :: ACCEPT AT(4,28)VALIDATE("JN")SIZE(1)BEEP:SKP$
440 IF SKP$="N" THEN 1000
450 DISPLAY AT(6,1):"FELDNAME ? (GERAET.NAME)"
460 ACCEPT AT(7,1)BEEP:DATEI$
470 OPEN #1:DATEI$,OUTPUT,SEQUENTIAL,INTERNAL,FIXED 80
480 FOR I=32 TO 143 :: PRINT #1:"",ASCII$(I-32):: NEXT I
490 CLOSE #1
1000 CALL CLEAR :: STOP :: END
2000 ! GENERIEREN DER ZEICHEN
2010 UHEX$="0123456789ABCDEF"
2020 HEX$=""
2025 CALL HCHAR(21,1,32,128):: CALL HCHAR(19,28,32)
2030 IF MTX=0 THEN GOTO 2100
2040 BS=5 :: BZE=20 :: GOSUB 5000
2050 BS=13 :: GOSUB 5000
2060 CALL CHAR(96,HEX$):: CALL MAGNIFY(4):: CALL SPRITE(#1,96,2,41,185)
2062 FOR I=0 TO 3 :: CALL CHARPAT(96+I,HEX$):: DISPLAY AT(21+I,3):HEX$ :: NEXT I
2070 GOTO 2200
2100 BS=5 :: BZE=12
2110 GOSUB 5000
2120 CALL CHAR(96,HEX$):: CALL MAGNIFY(2):: CALL SPRITE(#1,96,2,33,121)
2130 DISPLAY AT(21,3):HEX$
2200 DISPLAY AT(21,22):"WEITER" :: DISPLAY AT(22,22):" MIT" :: DISPLAY AT(23,22)
:"PROC'D"
2202 CALL KEY(5,T,S):: IF T<>12 THEN 2202
2208 CALL HCHAR(21,1,32,128):: CALL HCHAR(19,26,32):: DISPLAY AT(24,1):"CHARAKTE
R ABSPEICHERN (J/N)" :: ACCEPT AT(24,28)VALIDATE("JN")SIZE(1)BEEP:SKP$
2210 IF SKP$="J" THEN 2220 ELSE RETURN
2220 DISPLAY AT(24,1):"UNTER WELCHEM ZEICHEN ?" :: ACCEPT AT(24,25)SIZE(3)VALIDA
TE(DIGIT)BEEP:CHARNR
2230 IF CHARNR<32 OR CHARNR>143 THEN 2220
2240 IF MTX=1 THEN BACK=1 :: GOSUB 24000
2245 IF MTX=2 THEN MTX=1 :: GOTO 2200
2250 IF MTX=1 THEN 2270
2260 CALL CHARPAT(96,ASCII$(CHARNR-32)):: RETURN
2270 FOR I=0 TO 3
2280 CALL CHARPAT(96+I,ASCII$(CHARNR-32+I))
2290 NEXT I
2300 RETURN
3000 ! ZEICHEN SPIEGELN
3010 PS=5
3020 IF MTX=1 THEN PE=20 :: L=15 ELSE PE=12 :: L=7
3030 FOR I=5 TO L+5
3040 CALL GCHAR(I,PS,Z1):: CALL GCHAR(I,PE,Z2):: CALL HCHAR(I,PS,Z2):: CALL HCHA
R(I,PE,Z1)
3050 NEXT I
3060 PS=PS+1 :: PE=PE-1
3070 IF PE<PS THEN RETURN
3080 GOTO 3030
4000 ! DREHEN DES ZEICHENS
4010 FOR I=1 TO 16 :: FELD$(I)="" :: NEXT I
4020 IF MTX=1 THEN PS=5 :: PE=20 ELSE PS=5 :: PE=12
4030 IF SKIP=4 THEN 4200
```

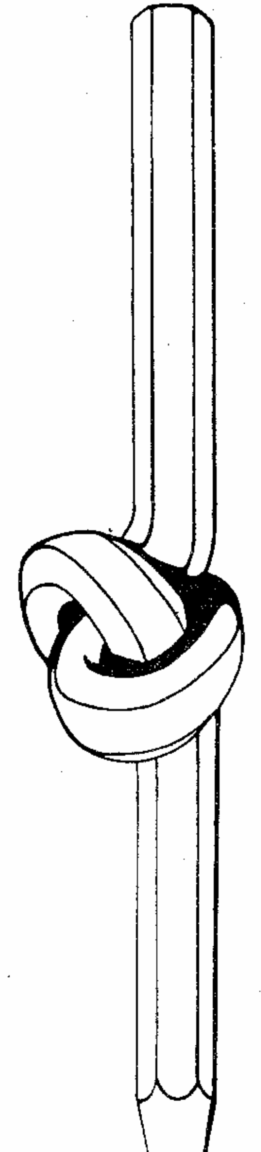
```
4040 FOR I=PS TO PE
4050 FOR K=PE TO PS STEP -1
4060 CALL GCHAR(K,I,Z1)
4070 FELD$(I-PS+1)=FELD$(I-PS+1)&CHR$(Z1)
4080 NEXT K
4090 NEXT I
4100 GOTO 4500
4200 J=0
4205 FOR I=PE TO PS STEP -1
4207 J=J+1
4210 FOR K=PS TO PE
4220 CALL GCHAR(K,I,Z1)
4230 FELD$(J)=FELD$(J)&CHR$(Z1)
4240 NEXT K
4250 NEXT I
4500 FOR I=PS TO PE
4510 FOR K=1 TO PE-PS+1
4520 CALL HCHAR(I,K+4,ASC(SEG$(FELD$(I-PS+1),K,1)))
4530 NEXT K
4540 NEXT I
4550 RETURN
5000 ! BINAER - HEXADEZIMAL - WANDLER
5010 FOR K=5 TO BZE
5020 FOR I=BS TO BS+7
5030 CALL GCHAR(K,I,F):: FELD(I-BS+1)=F-100
5040 NEXT I
5050 LOW=FELD(5)*8+FELD(6)*4+FELD(7)*2+FELD(8)+1
5060 HIGH=FELD(1)*8+FELD(2)*4+FELD(3)*2+FELD(4)+1
5070 HEX$=HEX$&SEG$(UHEX$,HIGH,1)&SEG$(UHEX$,LOW,1)
5080 NEXT K
5090 RETURN
20000 DISPLAY AT(23,1):" " :: DISPLAY AT(24,1):"WELCHEN CHARAKTER ?" :: ACCEPT A
T(24,21)SIZE(3)VALIDATE(DIGIT)BEEP:CHARNR
20005 BACK=0
20010 IF CHARNR<32 OR CHARNR>143 THEN 21000
20020 FOR I=1 TO 16 :: FELD$(I)="" :: NEXT I
20030 IF MTX=1 THEN 22000
20032 ASCI$=ASCII$(CHARNR-32)
20040 CALL ASCII(FELD$(),ASCI$)
20050 CALL CHARPRT(5,5,FELD$())
20060 CALL CHAR(96,ASCII$(CHARNR-32)):: GOTO 350
21000 DISPLAY AT(23,1):"NICHT MOEGLICH !!!" :: CALL SOUND(250,110,0)
21010 IF BACK=1 THEN MTX=2 :: RETURN ELSE 320
22000 DATA 5,5,13,5,5,13,13,13
22010 RESTORE 22000
22020 GOSUB 24000
22030 FOR I=0 TO 3
22035 FOR J=1 TO 16 :: FELD$(J)="" :: NEXT J
22040 ASCI$=ASCII$(CHARNR-32+I)
22050 CALL ASCII(FELD$(),ASCI$)
22060 READ P1,P2 :: CALL CHARPRT(P1,P2,FELD$())
22070 CALL CHAR(96+I,ASCII$(CHARNR-32+I))
22080 NEXT I
22090 GOTO 350
24000 ! KONTROLLE DER MAGNIFY(3) POSITIONEN
24010 FOR I=32 TO 140 STEP 4
24020 IF CHARNR=I THEN RETURN
24030 NEXT I
24040 GOTO 21000
25000 SUB M8 ! 8 X 8 MATRIX ENTSPR. 1 CHARAKTER
25010 FOR I=5 TO 12 :: CALL HCHAR(I,5,100,8):: NEXT I
25020 FOR I=1 TO 8 :: CALL HCHAR(4,4+I,I+48):: CALL VCHAR(4+I,4,I+48):: NEXT I
25030 CALL HCHAR(3,16,96):: FOR I=5 TO 6 :: CALL HCHAR(I,16,100,2):: NEXT I
```

```
25040 SUBEND
25050 SUB M16 ! 16 X 16 MATRIX ENTSPR. 4 CHARAKTEREN
25060 FOR I=1 TO 16 :: CALL HCHAR(4+I,5,100,16):: NEXT I
25070 CALL HCHAR(3,5,48,9):: CALL HCHAR(3,14,49,7)
25080 FOR I=1 TO 9 :: CALL HCHAR(4,4+I,I+48):: NEXT I
25090 FOR I=0 TO 6 :: CALL HCHAR(4,14+I,I+48):: NEXT I
25100 CALL VCHAR(5,3,48,9):: CALL VCHAR(14,3,49,7)
25110 FOR I=1 TO 9 :: CALL VCHAR(I+4,4,I+48):: NEXT I
25120 FOR I=0 TO 6 :: CALL VCHAR(I+14,4,I+48):: NEXT I
25130 CALL HCHAR(3,25,96):: CALL HCHAR(4,25,97):: CALL HCHAR(3,26,98):: CALL HCH
AR(4,26,99)
25140 FOR I=6 TO 9 :: CALL HCHAR(I,24,100,4):: NEXT I
25150 SUBEND
26000 SUB CHARPRT(Z,S,FELD$())
26010 FOR I=Z TO Z+7
26020 FOR K=S TO S+7
26030 CALL HCHAR(I,K,VAL(SEG$(FELD$(I-Z+1),K-S+1,1))+100)
26040 NEXT K
26050 NEXT I
26060 SUBEND
27000 SUB EDIT(XMAX,YMAX)! DEFINIEREN UND EDITIEREN DES CHARAKTERS
27002 CALL DELSPRITE(#1):: CALL MAGNIFY(1):: CALL SPRITE(#1,102,7,33,33)
27005 DISPLAY AT(23,1):"DRUECKEN SIE FCTN 'BACK' UM DIE DEFINITION ZU BEENDEN"
27010 X,Y=5
27020 CALL KEY(5,T,S):: IF S=0 THEN 27020
27040 IF T=48 OR T=49 THEN 27200
27050 IF T=88 THEN Y=Y+1
27060 IF T=83 THEN X=X-1
27070 IF T=68 THEN X=X+1
27080 IF T=69 THEN Y=Y-1
27090 IF T=15 THEN SUBEXIT
27100 IF X>XMAX THEN X=XMAX
27110 IF Y>YMAX THEN Y=YMAX
27120 IF X<5 THEN X=5
27130 IF Y<5 THEN Y=5
27140 CALL LOCATE(#1,(Y-1)*8+1,(X-1)*8+1)
27150 GOTO 27020
27200 CALL HCHAR(Y,X,T+52)
27210 X=X+1 :: IF X>XMAX THEN X=5 :: Y=Y+1
27220 IF Y>YMAX THEN Y=5
27230 GOTO 27140
27240 SUBEND
31999 SUB ASCII(FELD$(),HEX$)
32050 M=1
32060 C$=""
32070 H$=SEG$(HEX$,1,2)
32080 E=0
32090 FOR L=LEN(H$)TO 1 STEP -1
32100 B$=SEG$(H$,L,1)
32110 FOR J=0 TO 5
32120 IF ASC(B$)=65+J THEN A=10+J :: GOTO 32150
32130 NEXT J
32140 A=VAL(B$)
32150 D=D+A*16^E
32160 E=E+1
32170 NEXT L
32180 IF D=0 THEN FELD$(M)="00000000" :: GOTO 32240
32190 D=D/2
32200 IF D=INT(D)THEN C$="0"&C$ :: GOTO 32220
32210 C$="1"&C$ :: D=INT(D)
32220 IF INT(D)>0 THEN 32190
32230 FELD$(M)=C$ :: C$=""
```

```
32240 M=M+1
32250 HEX$=SEG$(HEX$,3,LEN(HEX$)-2)
32260 IF HEX$<>"" THEN 32070
32270 D=0
32280 FOR R=1 TO 8
32290 IF LEN(FELD$(R))<8 THEN FELD$(R)="0"&FELD$(R):: GOTO 32290
32300 NEXT R
32420 SUBEND
```

ASC II - SET Unterprogramm

```
32400 SUB ASCIISET(DATEI$)
32410 !
32420 ! Unterprogramm zur Umbelegung des Graphiksatzes
32430 !
32440 ! fuer II 99/4 A
32450 !
32460 ! (c) 1984 by Volker Becker
32470 !
32480 !
32482 CALL CLEAR
32485 FOR I=32 TO 143
32487 PRINT CHR$(I);" ";
32489 NEXT I
32490 OPEN #88:DATEI$,INPUT ,SEQUENTIAL,INTERNAL,FIXED 80
32500 FOR I=32 TO 143
32510 INPUT #88:GRAPH$,ASCII$
32520 CALL CHAR(I,ASCII$)
32530 NEXT I
32540 CLOSE #88
32550 SUBEND
```



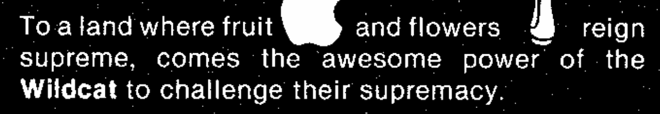
ASC II - PAT Unterprogramm

```
32000 SUB ASCIIPAT(DATEI$)
32005 !
32010 ! copieren eines Codefeldes aus einem bestehenden Programm
32015 !
32020 ! (c) 1984 by Volker Becker
32025 !
32030 ! fuer TI 99/4 A
32035 !
32040 !
32045 OPEN #88:DATEI$,OUTPUT,SEQUENTIAL,INTERNAL,FIXED 80
32050 FOR I=32 TO 143
32055 CALL CHARPAT(I,HEX$)
32060 PRINT #88:"",HEX$
32065 NEXT I
32070 CLOSE #88
32075 SUBEND
```

Achtung: Bei dem Programm ASCII-DEF kann es bei einer falschen Eingabe oder bei einer vollen Diskette am Programmende zu einer Fehlermeldung kommen (I/0 ERROR). Wenn dies geschieht, so kann man die im Rechner befindliche Datei retten, da sonst die ganze Definition umsonst gewesen wäre.
Dazu gibt man als Befehl, nachdem die Fehlermeldung gekommen ist, die nachfolgende Zeile ein:
OPEN 7: „Gerät. Name", OUTPUT, SEQUENTIAL, INTERNAL, FIXED 80 : : FOR I=32 TO 143 : : PRINT 7: „„,ASCII$ (I-32) : : NEXT I
Danach ENTER drücken. Der Rechner rettet jetzt die Datei auf den entsprechenden externen Speicher. (Siehe Programm ASCIIDEF Zeilen 470 und 480).
Dies geht jedoch nur, wenn noch kein anderer Befehl nach der Fehlermeldung gegeben wurde.

WILDCAT™

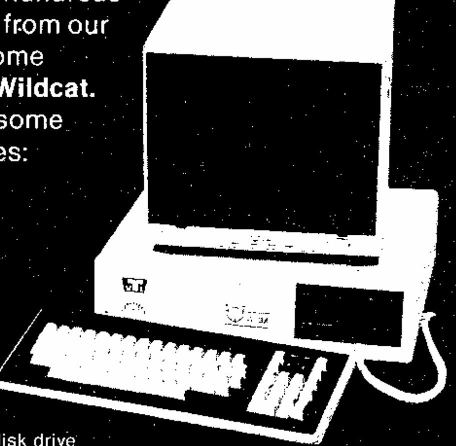
Looks Like an IBM™ Works Like an Apple™



To a land where fruit and flowers reign supreme, comes the awesome power of the **Wildcat** to challenge their supremacy.

Wildcat is a sleek styled mobile computer designed for your vehicle, boat or aircraft. It looks like an IBM PC™ and is fully software compatible with the Apple II™ product line at a list price that would put a smile on the face of the most frugal computer buyer.

But price isn't everything. All those features that would cost you hundreds of dollars extra from our competitors, come standard with **Wildcat.** Let's compare some of these features:



	Apple IIe	Wildcat
Detachable keyboard	No	STD
Full numeric key pad	Option	STD
Full functions keys	No	41
Built in disk controller	No	STD
Parallel printer port	No	STD
RS 232 serial port	No	STD
Game port	1	2
RGB video out	Option	STD
Composite video	STD	STD
RF video for TV	Option	STD
CP/M	Option	STD
Hi Res graphics (6 color)	STD	STD
Low Res graphics (16 color)	STD	STD
64KB memory	STD	STD
Half high disk drives	No	STD
Converters for vehicles, boats, and aircraft	No	Option
Aluminum carrying case	No	Option
List price	$1940*	$1099.00*

For more information on the all new **Wildcat**, see your local computer dealer or call or write:



COMPUTER AND PERIPHERAL PRODUCTS
1530 S. Sinclair
Anaheim, CA 92806
(714) 978-9820

*Computer plus one disk drive

Warning: This equipment is exempt from compliance with FCC testing requirements pursuant to 47 CFR 15.801 (c) (1)
Operation of this equipment in a residential area may cause interference.

IBM is the registered trademark of International Business Machines Corp.
Apple is the registered trademark of Apple Computer, Inc.
Apple II is the trademark of Apple Computer, Inc.

CIRCLE 181 ON READER SERVICE CARD

ZX Spectrum

Mathematik-programm

Mathematikprogramm

Dieses Programm für den ZX-Spectrum paßt mit einer Länge von ca. 7,43 Kb noch in die 16-Kb-Version. Und weist dennoch folgende Besonderheiten auf:

Alle Rechenarten sind aus der höheren Mathematik und daher für Schüler der 12. bis 13. Klasse sowie für Studenten der Naturwissenschaften interessant.

Es weist folgende 5 Themen auf:

a) Integralrechnung
b) Differentialrechnung
c) Newtonsche Näherung
d) Monotonie
e) Plotten von Funktionen

Für das Zeichnen der Funktionen brauchen keine Unstetigkeitsstellen eingegeben werden. D. h., jede beliebige Funktion mit beliebig vielen Definitionslücken kann dargestellt und auf Wunsch auf dem Drucker ausgegeben werden. Die Funktion ist vertikal beliebig vergrößerbar. Ausschnittsvergrößerungen von interessanten Stellen können daher besonders untersucht werden.

Alle Unterprogramme werden über Menuekarte angesteuert. Das Programm hat noch einen Instruktionenteil, der evtl. Probleme weitgehendst beseitigt.

Da das Programm modular aufgebaut ist, können auch einzelne Unterprogramme allein benutzt werden, so daß ein Abtippen des gesamten Programms nicht unbedingt notwendig ist.

Beim Integrieren brauchen keine Nullstellen vorher berechnet zu werden, da das Programm negative Vorzeichen automatisch berücksichtigt.




```
  1 REM MATHE 1 VERSION 9
  2 INK 7: BRIGHT 1
  3 BORDER 0
  4 PAPER 0: CLS
  5  PRINT AT 0,4;"  COPYRIGHT BY UWE HAFERLAND"
  6  FOR U=1 TO 20
  7  PRINT AT U,4;CHR$ 143
  8  PRINT AT U,18;CHR$ 143
  9  NEXT U
 10  FOR K=1 TO 21
 11  PRINT AT K,20;CHR$ 143
 12  PRINT AT K,31;CHR$ 143
 13  NEXT K
 14  FOR W=4 TO 18
 15  PRINT AT 21,W;CHR$ 143
 16  NEXT W
 17  FOR H=20 TO 31
 18  PRINT AT 11,H;CHR$ 143
 19  NEXT H
 20 PRINT #0, FLASH 1;AT 0,1;"MATHE 1 VERSION 9;83/84":  PAUSE 200: CLS : POKE
23658,8
 21 CLS : PRINT "DIESES PROGRAMM BEINHALTET FOL- GENDE RECHENARTEN:
                                DIFFERENTIALRECHNUNG          D
                 INTEGRALRECHNUNG                  I
 GRAPHISCHE DARSTELLUNG VON     FUNKTIONEN                    F
                 MONOTONIE                         D
 NULLSTELLENBERECHNUNG MIT      NEWTONSCHER NAEHERUNG         N
                 ALLGEMEINE INSTRUKTIONEN          M"

 22 PRINT AT 19,0;"DRUECKEN SIE NUN EINES DER       ZEICHEN AM RECHTEN BILDSCHIR
M-  RAND ZUR WAHL IHRER RECHENART"
 24 IF INKEY$="I" THEN  GO TO 400
 25 IF INKEY$="F" THEN  GO TO 300
 26 IF INKEY$="D" THEN  GO TO 200
```

```
  27 IF INKEY$="N" THEN  GO TO 800
  28 IF INKEY$="M" THEN  GO TO 600
  29 IF INKEY$="M" THEN  GO TO 600
  30 GO TO 24
 200 REM DIFFERENTIALRECHNUNG
 202 CLS : PRINT AT 1,0;"DIFFERENTIALRECHNUNG/MONOTONIE"
 204 PRINT AT 4,0;"BEI ERSCHEINEN DER FEHLERMELDUNG NR.6 NUMBER TOO BIG IST DIE
   FUNKTION NICHT STETIG UND DAHER IN DIESER FORM NICHT            DIFFERENZIERB
AR!"
 206 INPUT "GEBEN SIE NUN DIE FUNKTION EIN  MIT X ALS ARGUMENT,ABER OHNE DIEABHA
ENGIGE VARIABLE Y:             ";A$
 210 INPUT "GEBEN SIE X0 EIN",X
 212  LET A=.1E-4
 214  LET B=1
 216  DIM Y(3)
 218  GO SUB 244
 220  LET X=X+A
 222  GO SUB 242
 224  LET X=X-2*A
 226  GO SUB 242
 228  LET C=(Y(2)-Y(3))/(2*A)
 230  CLS
 232  PRINT "DIE STEIGUNG BETRAEGT AN DIESER STELLE: "; C
 233 IF ABS C>1E6 THEN  PRINT AT 0,0;"AN DIESER STELLE LIEGT           WAHRSCHEIN
LICH EINE UNENDLICH-  KEITSSTELLE VOR,DIE FUNKTION ISTNICHT DIFFERENZIERBAR!"
 234  IF C>0 THEN  PRINT "DIE FUNKTION  IST STRENG MONOTON STEIGEND AN DER STELL
E X=";X+A
 236  IF C<0 THEN  PRINT "DIE FUNKTION IST STRENG MONOTON FALLEND AN DER STELLE
X=";X+A
 240  GO TO 853
 242  LET B=B+1
 243 LET Y(B)=VAL A$
 244  RETURN
 300 REM FUNKTIONEN
 302  CLS
 303 PRINT AT 0,12;"FUNKTIONEN"
 304 INPUT "GEBEN SIE NUN DIE FUNKTION EIN  MIT X ALS ARGUMENT,ABER OHNE DIEABHA
ENGIGE VARIABLE Y:             ";A$
 307 LET KONTROLLE=0
 308  LET D=1
 312  INPUT "GEBEN SIE DEN DEFINITIONSBEREICH VON...BIS...EIN,SOWIE DIE      SCH
RITTFOLGE VON X:",A,B,C
 313 INPUT "VERGROESZERUNG: ";G
 314  LET FLAG=0
 316 LET P=880/G
 317 LET U=P-INT P
 318 IF U>=.5 THEN  LET P=INT P+1
 319 IF U<0.5 THEN  LET P=INT P
 320 LET P=0.1*P
 321 LET I=A
 322  IF ABS A>=ABS B THEN  LET E=ABS A
 323 CLS
 324 IF ABS A<ABS B THEN  LET E=ABS B
 326  IF A=0 THEN  LET E=ABS B
 328  IF B=0 THEN  LET E=ABS A
 332  IF A>=0 THEN  LET FLAG=2
 334  FOR K=0 TO 31
 336  PRINT AT 10,K;"."
 337  NEXT K
 338 OVER 1: PRINT AT 11,0;"A";AT 11,31;"B"
 339 IF A<0 THEN  PRINT AT 0,16;"P";AT 21,16;"Q"
 340 IF A>=0 THEN  PRINT AT 0,1;"P";AT 21,1;"Q"
 341  FOR L=0 TO 21
 342  IF A>=0  THEN  PRINT AT L,0;"."
 344 IF A<0 THEN  PRINT AT L,15;"."
 346  NEXT L
```

```
 348  NEXT K
 355 IF KONTROLLE=1 THEN  LET A=VAL "X"
 356  FOR X=A TO B STEP C
 357 LET Y=VAL A$
 360  IF ABS (Y*G/D+88)>175 OR (Y*G/D+88)<0 THEN  GO TO 372
 362  IF FLAG=2  THEN  GO TO 376
 364  PLOT (X-C)*127/E+128,Y*G/D+88
 366  NEXT X
 368  GO TO 380
 372 LET KONTROLLE=1: LET X=X+0.1
 374  GO TO 355
 376  PLOT (X-C)*255/E,Y*G/D+88
 378  NEXT X
 380  PRINT #0;AT 0,0;"A=";I;" ";"B=";B;" ";"P=";P
 382  PRINT #0;AT 1,0;"Q=";-P;" 0=NEUE AUFGABE 1=COPIE"
 383 IF INKEY$="1" THEN  GO TO 390
 384  IF INKEY$="0"  THEN  GO TO 21
 385 GO TO 383
 390 LPRINT "Y=";A$;TAB 79;" ": COPY : LPRINT ;"
                                               ";"A=";I;"  B=";B;"  P=";P;"  Q
=";-P: GO TO 21
 400 REM INTEGRALE
 401 CLS : PRINT AT 0,13;"INTEGRALE"
 403 INPUT "GEBEN SIE NUN DIE FUNKTION EIN  MIT X ALS ARGUMENT,ABER OHNE DIEABHA
ENGIGE VARIABLE Y:          ";A$
 406 INPUT "UNTERE GRENZE",D,"OBERE GRENZE",E,"ANZAHL DER SUBINTERVALLE",F
 407 CLS
 408 IF F>=40 THEN  PRINT "BITTE WARTEN!"
 409 LET B=(E-D)/2/F
 412 LET A=0: LET X=D: GO SUB 436
 415 FOR G=1 TO F
 418 LET A=Y+A: LET X=X+B: GO SUB 436
 421 LET A=Y*4+A: LET X=X+B: GO SUB 436
 424 LET A=Y+A
 427 NEXT G
 430 LET C=A*B/3
 431 CLS
 432 BEEP .5,30 : BEEP 0.5,27
 433 PRINT "DIE FLAECHE BETRAEGT",C: GO TO 853
 436 LET Y=VAL A$
 439 LET y=ABS y
 442 RETURN
 600 CLS : PRINT AT 0,3;"ALLGEMEINE INSTRUKTIONEN"
 601 PRINT AT 2,0;"1. BEI EINGABE DER FUNKTIONEN   IST X IMMER DIE VARIABLE.
  .2. BEDENKEN SIE,DASS DER RECHNERBEI TRIGONOMETRISCHEN FUNKTIONENIMMER DIE VAR
IABLE ALS BOGENMASSBETRACHTET.WUENSCHEN SIE DIE    EINHEIT IN GRAD,SO TIPPEN SIE
   FUER DIE VARIABLE DEN WINKEL IN GRAD EIN,UND MULTIPLIZIEREN     DIESE MIT PI/
180,WOBEI DIESER   FAKTOR MIT DEM ARGUMENT IN      KLAMMERN STEHEN MUSS!ZUM BEIS
P.:SIN(90*PI/180)MIT 90 GRAD ALS   VARIABLE IN DEM BEISPIEL.       3. BEI AUFTRE
TEN DER FEHLER-    MELDUNG NR.6,NUMBER TOO BIG,IST DIE EINGEGEBENE FUNKTION  NIC
HT STETIG,DAHER AUCH MEISTENS NICHT LOESBAR."
 602 PRINT AT 20,9;"DAS FUNKTIONENPROGRAMM DARF NIE MIT EINER UNSTETIGKEITSSTELL
E BEGINNEN!!"
 603 PRINT #0,AT 1,0;"DRUECKEN SIE Z BEI BEENDIGUNG DIESER SEITE:"
 604 IF INKEY$="Z" THEN  CLS : IF INKEY$="Z" THEN  PRINT "4.WIRD DIE VERGROESSER
UNG ZU    GROSS GEWAEHLT,SO WIRD DIE      FUNKTION NUR AUF EINEM TEIL DER X-ACHS
E LIEGEN.WAEHLEN SIE DANN EINE KLEINERE VERGROESSERUNG,SO ERSTRECKT SICH NUN DER
 GRAPH    UEBER DIE GANZE ABSZISSE.
                                 DRUECKEN SIE W BEI BEENDIGUNG!"
 610 IF INKEY$="W" THEN  GO TO 21
 612 GO TO 604
 800 REM NEWTONSCHE NAEHERUNG
 801 LET SIGNAL=0
 802 CLS : PRINT AT 0,0;"NULLSTELLENBESTIMMUNG MIT        NEWTONSCHER NAEHERUNG
  "
 803 INPUT "GEBEN SIE NUN DIE FUNKTION EIN  MIT X ALS ARGUMENT,ABER OHNE DIEABHA
```

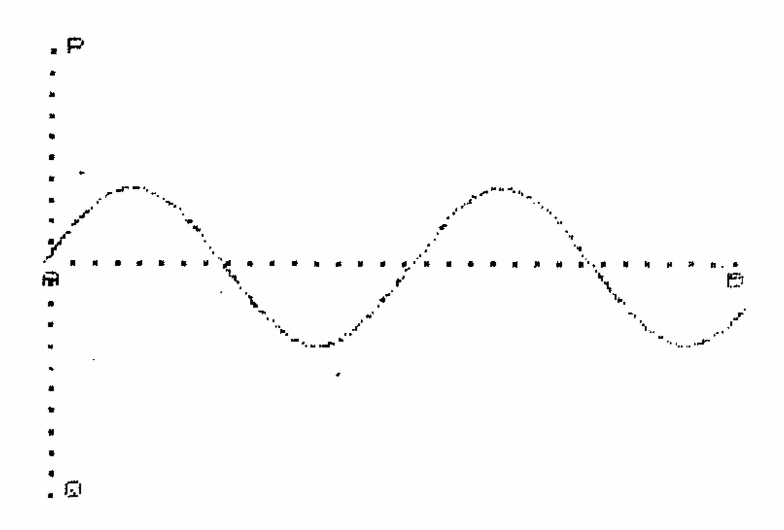
```
ENGIGE VARIABLE Y.HANDELT ESSICH UM EINE GEBROCHEN RATIONALEFUNKTION,GEBEN SIE N
UR DEN       ZAEHLER EIN,UND PRUEFEN SIE      SPAETER DURCH EINGABE DES       NENN
ERS,OB DIESER EINE NULL-    STELLE BEI X=0 HAT.IST DIES DER FALL,SO MUESSEN SIE
GRENZWERT-  BESTIMMUNGEN MACHEN.IM ANDEREN  FALL IST DIE NULLSTELLE DES      ZAEH
LERS GLEICH DIE NULLSTELLE  DER GESAMTEN FUNKTION.";A$
 804 LET M=0: DIM Y(3): OVER 0
 805  LET B=0: LET F=999999999: LET A=0.00001
 806 INPUT "GEBEN SIE EINE GESCHAETZTE       NULLSTELLE EIN.VERMUTEN SIE      WEIT
ERE,SO STARTEN SIE DIESES    PROGRAMM ERNEUT,ALLERDINGS MIT  EINEM ANDEREN SCHAET
ZWERT:",E
 807 INPUT "GEBEN SIE EIN,AB WELCHER        NAEHERUNG DIE RECHNUNG GESTOPPT WERD
EN SOLL: ";N
 808 LET X=VAL "E"
 810 LET M=M+1
 811 IF M=100 THEN  PRINT AT 19,0;"KEINE NULLSTELLE GEFUNDEN!       ": PAUSE 450:
 GO TO 21
 812 PRINT AT 18,0;"NAEHERUNG NR.";M
 814  LET B=1
 818  GO SUB 850
 821  LET X=X+A
 822 LET B=2
 824  GO SUB 848
 825 LET B=3
 827  LET X=X-2*A
 830  GO SUB 848
 833  LET C=(Y(2)-Y(3))/(2*A)
 837 LET F=VAL "E"
 838 LET E=X-Y(1)/C: IF ABS E>=1E5 THEN  PRINT "WAHRSCHEINLICH KEINE NULLSTELLE
 VORHANDEN!": PAUSE 300: IF ABS E>=1E5 THEN  GO TO 21
 839 PRINT AT 19,0;"ZWISCHENERGEBNISS:";E
 840 IF ABS (F-E)>N THEN  GO TO 808
 841 IF ABS (F-E)<=N THEN  GO TO 852
 842 IF (F-E)<=N THEN  STOP
 843>LET F=VAL "E"
 845 STOP
 850 LET Y(B)=VAL A$
 851  RETURN
 852 PRINT AT 19,0;"DIE NULLSTELLE LIEGT BEI X=";E;"      "
 853 IF INKEY$<>"" THEN  GO TO 21
 854 GO TO 853
```

```
Y=2*(X*X)+X-2

P

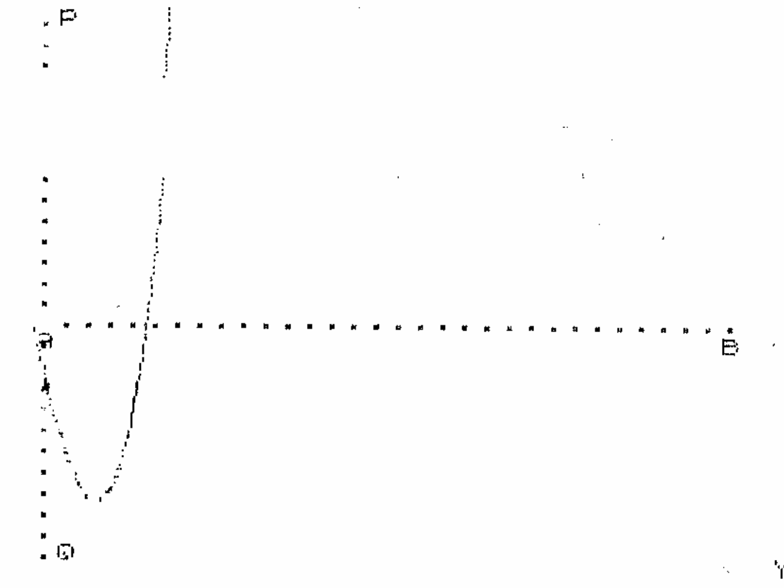
A                                              B

Q       A=1   B=5   P=88   Q=-88
Y=SIN X
```



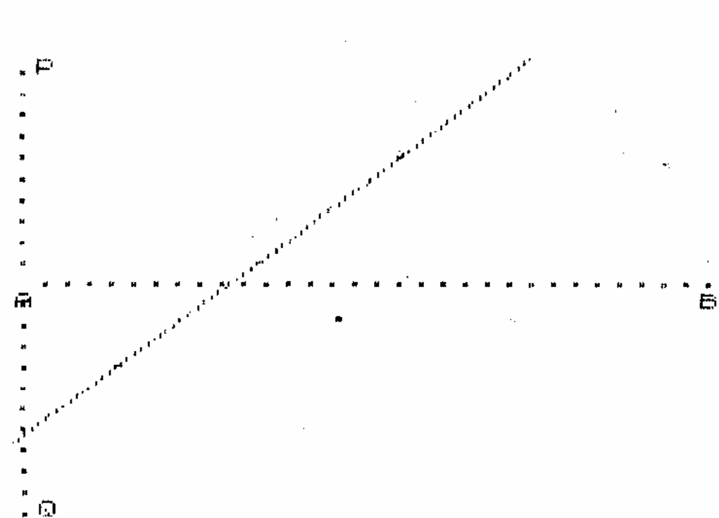
A=0 B=12 P=2.9 Q=-2.9

Y=(X*X*X)-8*X+(X*X)



A=0 B=15 P=8.8 Q=-8.8

Y=2*X-6



A=0 B=10 P=8.8 Q=-8.8

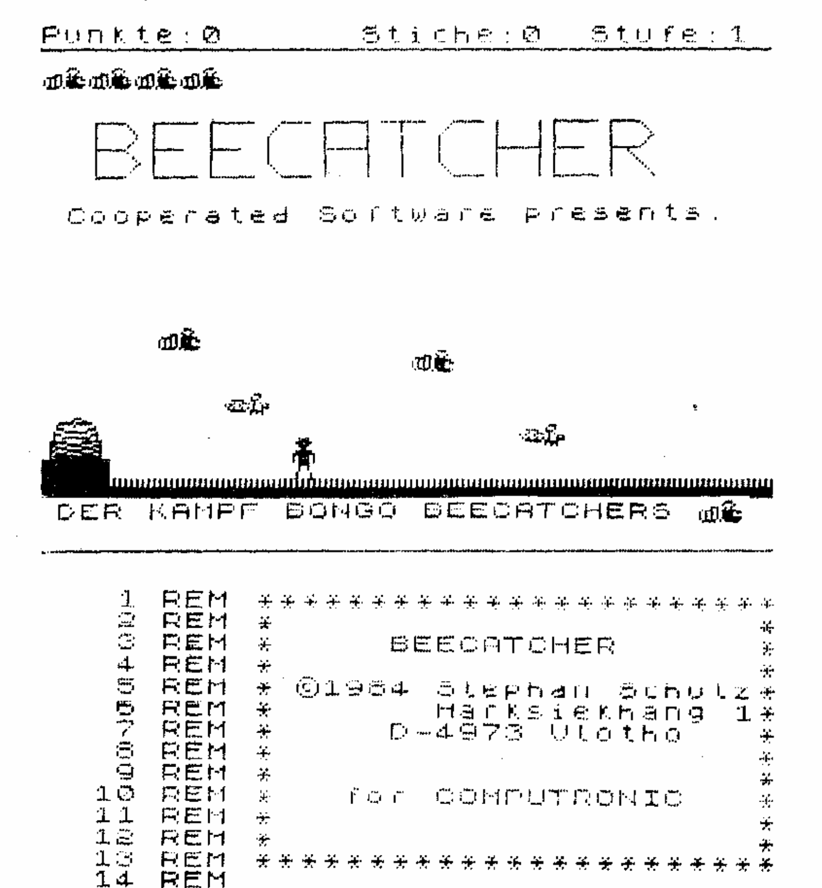
ZX Spectrum

Bongo Beecatcher

Bongo Beecatcher
für ZX-Spectrum 48 K
In diesem Spiel geht es darum, daß die Hauptperson, Bongo Beecatcher, die Bienen, die aus Ihrem Korb geflohen sind, wieder einfangen muß. Dieses tut er, indem er in die Höhe springt und versucht, die Biene zu ergreifen. Jedoch Vorsicht, einige Hornissen, die zwischen der Biene und dem Boden fliegen, versuchen, Ihn daran zu hindern. Stößt Bongo mit einer von ihnen zusammen, so wird er von Ihr gestochen. Vier dieser Stiche kann Bongo aushalten, beim fünften muß er sterben.
Die Schwierigkeit bei diesem Spiel erhöht sich mit der Zeit durch zwei Faktoren. Zum ersten nimmt die Anzahl der Hornissen immer weiter zu (von einer einzigen am Anfang bis zu acht ab Stufe vier), zum zweiten fliegt die Biene auf immer unregelmäßigeren Bahnen.
Das Spiel braucht in der Form, in der es abgedruckt ist, mit allen Variablen etwa 10.7 Kbyte und ist somit nur auf der 48K-Version lauffähig. Durch weglassen aller REMs und der Spielanleitung (Zeilen 7500-7540) kann es jedoch soweit gekürzt werden, daß es auch auf dem 16K-Spectrum läuft. Viel Spaß!

Programmerklärung nach Zeilennummern

Zeilen:	Inhalt:
1-14	REMs für Vorspann
15	Setzen von PAPER, BORDER und INK, löschen des Bildschirms, zwischendurch Kurzvorspann
20-110	UDGs, DATAs und Leseschleife
200-250	Definition der Variablen, die pro Spiel nur einmal auf Anfangswert gesetzt werden
300-360	DATAs, die die verschiedenen Flughöhen der Biene in verschiedenen Stufen angeben




```
   1 REM ************************
   2 REM *                      *
   3 REM *      BEECATCHER      *
   4 REM *                      *
   5 REM * ©1984 Stephan Schulz *
   6 REM *        Marksiekhang 1*
   7 REM *      D-4973 Vlotho   *
   8 REM *                      *
   9 REM *                      *
  10 REM *    for COMPUTRONIC   *
  11 REM *                      *
  12 REM *                      *
  13 REM ************************
  14 REM
  15 BORDER 0: INK 7: PAPER 0: CLS : FOR f=1 TO 250: PRINT AT 10,2; INK (7*RND);"Cooperated Software presents:";: NEXT f: CLS
  20 REM UDGS
  30 RESTORE : FOR f=0 TO 119: READ a: POKE USR "a"+f,a: NEXT f
  40 DATA 0,0,62,83,164,82,63,0,56,72,112,171,37,167,124,84
  50 DATA 112,248,240,112,32,112,240,248,248,248,240,80,80,136,136,204
  60 DATA 14,31,30,14,4,14,31,31,31,31,30,4,4,4,4,6
  70 DATA 14,31,15,14,4,14,31,31,31,31,15,10,10,17,17,51
  80 DATA 112,248,120,112,32,112,248,248,248,248,120,32,32,32,32,96
  90 DATA 60,126,102,60,24,128,255,189,189,189,60,36,36,66,66,195
 100 DATA 0,0,7,61,85,85,85,62,56,68,40,126,123,120,127,190
 110 DATA 85,85,85,255,255,255,255,255
```

ZX Spectrum

400	Einlesen dieser Werte in a(1–31) in jeder Stufe
410–420	Definition von c$ in den verschiedenen Stufen
500–560	Bildschirmaufbau
1000–1100	Hauptschleife
1020–1030	Steuerung rechts/links
1040	PRINTen der Biene
1050	Sprung zur Bewegung der Hornissen
1060–1070	Verzweigung zum Sprung, falls nötig
1100	PRINTen der Figur, Tonausgabe, Ende der Hauptschleife, löschen der Biene an der letzten Position, erneuter Sprung zum Beginn der Hauptschleife
2000–2020	PRINTen der Hornissen
2200–2300	Sprung der Figur mit Kontrolle, ob die Figur mit einer Hornisse oder einer Biene zusammengestoßen ist
2500–2520	Von Hornisse gestochen, mit Überprüfung, ob das Spiel zu Ende ist
2600–2630	Next Level mit Überprüfung, ob Spieler gewonnen hat
2800–2900	Biene gefangen, Überprüfung, ob next Level
3000	Spieler hat gewonnen
5000	Spieler hat verloren
6000	Abfrage „noch einmal?“
7000–7210	Anfangsgraphik
7500–7550	Anleitung und Sprung zum Spielanfang



```
 150 GO TO 7000
 200 REM  VARIABLEN
 210 LET a=19: LET b=10: LET c=3
2
 230 LET f$="
"
 240 LET d$=""
 250 LET bee=4: LET sti=0: LET p
u=0: LET st=1: LET e$=""
 300 REM  DATAs fuer unterschied
liche Stufen
 310 DATA 14,14,14,14,14,14,14,1
4,14,14,14,14,14,14,14,14,14,14,
14,14,14,14,14,14,14,14,14,14,14
,14,14
 320 DATA 13,14,14,13,14,14,13,1
4,14,13,14,14,13,14,14,13,14,14,
13,14,14,13,14,14,13,14,14,13,14
,14,13
 330 DATA 13,14,13,14,13,14,13,1
4,13,14,13,14,13,14,13,14,13,14,
13,14,13,14,13,14,13,14,13,14,13
,14,13
 340 DATA 12,13,14,13,12,13,14,1
3,12,13,14,13,12,13,14,13,12,13,
14,13,12,13,14,13,12,13,14,13,12
,13,14
 350 DATA 10,11,11,12,13,14,14,1
3,12,11,10,09,10,11,12,13,14,13,
13,12,11,10,10,09,09,10,11,12,11
,11,10
 360 DATA 06,07,08,08,08,07,08,0
9,10,11,11,12,12,11,12,13,14,13,
12,11,10,09,09,08,09,10,09,08,07
,07,06
 400 DIM a(31): RESTORE 300+10*s
t: FOR f=1 TO 31: READ a(f): NEX
T f
 410 LET hst=st: IF hst>4 THEN L
ET hst=4
 420 LET c$="": FOR f=1 TO 2↑hst
: LET c$=c$+f$(1 TO 32/(2↑(hst-1
))): NEXT f
 500 REM  BILDSCHIRMAUFBAU
 510 CLS : FOR f=2 TO 31: PRINT
AT 21,f; INK 4;"▀";: NEXT f
 520 PRINT INK 4;AT 21,0;"██";A
T 20,0;"██"
 530 INK 6: PLOT 3,16: DRAW 17,0
: DRAW 0,5: DRAW -17,0,PI: DRAW
0,-5: PLOT 3,18: DRAW 17,0,.7: P
LOT 3,20: DRAW 17,0,0.7: PLOT 3,
22: DRAW 17,0,.7
 540 PLOT 5,24: DRAW 13,0,.7: PL
OT 6,26: DRAW 11,0,.7: PLOT 8,28
: DRAW 7,0,.7
 550 INK 7: PRINT AT 0,0;"Punkte
:0         Stiche:0  Stufe:1 ";: PL
OT 0,166: DRAW 255,0
 560 PRINT AT 2,0; INK 6;e$; INK
 7;AT 0,7;pu;AT 0,21;sti;AT 0,30
;st
1000 REM  HAUPTSCHLEIFE
1010 FOR f=0 TO 29 STEP .5: LET
a$="      ": LET b$=a$
1020 IF IN 65278=189 AND b>3 THE
N LET b=b-1: LET a$="      ": LET
 b$="      "
1030 IF IN 65278=187 AND b<29 TH
EN LET b=b+1: LET a$="      ": LE
T b$="      "
1040 PRINT AT a(f+1),f; INK 6; B
RIGHT 1;"  ";AT a(f+1)+1,f;"
";AT a(f+1)-1,f;"   ";
1050 GO TO 2000
```

```
1060 IF d$<>" " THEN IF IN 32766
=190 THEN LET d$=" ": LET x=0
1070 IF d$=" " THEN GO TO 2200
1100 BRIGHT 0: INK 2: PRINT AT a
+1,b;a$(3 TO 5);AT a,b;a$(1 TO 3
);: BEEP .01,0: PRINT AT a+1,b;b
$(3 TO 5);AT a,b;b$(1 TO 3): NEX
T f: PRINT AT a(f+1),f;"  ";AT
a(f+1)-1,f;"  ",AT a(f+1)+1,f;"
  ";: GO TO 1010: REM      END
E HAUPTSCHLEIFE
2000 REM  HORNISSEN
2010>PRINT AT 17,0; INK 6;c$(c T
O c+31): LET c=c-1: IF c=1 THEN
LET c=32
2020 GO TO 1051
2200 REM  SPRUNG
2210 LET x=x+1: IF x<6 THEN LET
a=a-1: PRINT AT a+2,b;"   "
2220 IF x>7 THEN LET a=a+1: PRIN
T AT a-1,b;"   "
2230 IF a=19 THEN LET d$=""
2240 IF (a=17 OR a=18) AND (c$(c
+1+b)<>" " OR c$(c+2+b)<>" ") TH
EN GO TO 2500
2250 IF a=14 THEN PRINT AT a,b+1
;"  "; INK 6; BRIGHT 1;AT a(f+1
),f;"  ";: IF SCREEN$ (a,b+1)<>
"  " THEN GO TO 2800
2300 GO TO 1071
2500 REM  VON HORNISSE GESTOCHEN
2510 LET sti=sti+1: PRINT AT 0,2
1; INK 7;sti;AT a,b;"   ";AT a+1
,b;"   ": IF sti>4 THEN GO TO 50
00
2520 LET d$="": LET a=19: BEEP .
1,0: BEEP .1,10: BEEP .1,20: GO
TO 1071
2600 REM  NEXT LEVEL
2610 LET st=st+1: PRINT AT 9,10;
"          ";AT 10,10;" Next L
evel ";AT 11,10;"          ";:
 BEEP .1,10: BEEP .1,20: BEEP .1
,30: BEEP .1,40: BEEP .1,50: PRI
NT AT 0,30;st
2620 LET bee=4: IF st>6 THEN GO

TO 3000
2630 GO TO 400
2800 REM  BIENE GEFANGEN
2810 PRINT AT a(f+1),f;"  ": LE
T bee=bee-1: LET pu=pu+10*st: BE
EP .1,60: BEEP .1,40: BEEP .1,-1
0: PRINT AT a,b-2;"   ";AT a+1,
b;"   ": FOR g=b TO 3 STEP -1: P
RINT INK 6;AT 19,g;"  ";AT 20,g
;"  ";: BEEP .01,g: BEEP .01,-g
: PRINT AT 19,g;"  ";AT 20,g;"
  ";: NEXT g: LET b=3
2820 LET a=19: PRINT AT 0,7; INK
 7;pu,AT 2,0;"          ": IF bee>0
 THEN PRINT AT 2,0; INK 6;e$(1 T
O (2*bee));"  "
2890 LET d$="": IF bee<0 THEN GO
 TO 2600
2900 GO TO 1071
3000 REM  GEWONNEN
3010 PAUSE 100: FOR f=-50 TO 50:
 BEEP .01,f: BEEP .01,-f: NEXT f
: PRINT AT 10,11; FLASH 1;">>BRA
VO<<";: FOR f=1 TO 30: FOR g=0 T
3010>PAUSE 100: FOR f=-50 TO 50:
 BEEP .01,f: BEEP .01,-f: NEXT f
: PRINT AT 10,11; FLASH 1;">>BRA
VO<<";: FOR f=1 TO 30: FOR g=0 T
O 7: BORDER g: NEXT g: NEXT f: C
LS
```

```
3020 BORDER 0: PAPER 0: CLS : PR
INT "Du hast es geschaft !!!!
     Bongo Beecatcher konnte all
e    entflohenen Bienen wieder e
in-  fangen.Eine hervorragende
     Leistung......": GO TO 6000
5000 REM  VERLOREN
5010 BEEP .2,-30: PAPER 0: INK 7
 : BORDER 0: CLS : POKE 23692,255
 : FOR f=0 TO 20: PRINT BRIGHT 0;
">>>>>>>>>>> VERLOREN <<<<<<<<<
<", BRIGHT 1;">>>>>>>>>>> VERLOR
EN <<<<<<<<<<<";: NEXT f
5020 CLS : PRINT "Du hast es nic
ht geschafft !!!  Bongo Beecatch
er ist von den    boesen Horniss
en erstochen      worden.
                  UND WER IST SC
HULD ?????
                            DU": P
RINT : PRINT : PRINT : PRINT TAB
 4; FLASH 1;"Moerder...."
6000 PRINT #1;"nocheinmal ?(j/n)
"
6010 IF INKEY$="j" OR INKEY$="J"
 THEN RUN 200
6020 IF INKEY$="n" OR INKEY$="N"
 THEN STOP
6030 GO TO 6010
7000 REM  ANFANGSGRAPHIK
7010 PLOT 10,160: DRAW 10,0: DRA
W 5,-5: DRAW -5,-5: DRAW -10,0:
DRAW 10,0: DRAW 5,-5: DRAW -5,-5
 : DRAW -10,0: DRAW 0,20
7020 PLOT 45,160: DRAW -15,0: DR
AW 0,-10: DRAW 10,0: DRAW -10,0:
 DRAW 0,-10: DRAW 15,0
7030 PLOT 65,160: DRAW -15,0: DR
AW 0,-10: DRAW 10,0: DRAW -10,0:
 DRAW 0,-10: DRAW 15,0
7040 PLOT 85,160: DRAW -10,0: DR
AW -5,-5: DRAW 0,-10: DRAW 5,-5:
 DRAW 10,0
7050 PLOT 90,140: DRAW 0,20: DRA
W 15,0: DRAW 0,-10: DRAW -15,0:
DRAW 15,0: DRAW 0,-10
7060 PLOT 110,160: DRAW 15,0: DR
AW -8,0: DRAW 0,-20
7070 PLOT 145,160: DRAW -10,0: D
RAW -5,-5: DRAW 0,-10: DRAW 5,-5
 : DRAW 10,0
7080 PLOT 150,160: DRAW 0,-20: D
RAW 0,10: DRAW 15,0: DRAW 0,-10:
 DRAW 0,20
7090 PLOT 185,160: DRAW -15,0: D
RAW 0,-10: DRAW 10,0: DRAW -10,0
 : DRAW 0,-10: DRAW 15,0
7100 PLOT 190,160: DRAW 10,0: DR
AW 5,-5: DRAW -5,-5: DRAW -10,0:
 DRAW 0,10: DRAW 0,-20: DRAW 0,1
0: DRAW 5,0: DRAW 10,-10
7200 LET a$="  DER KAMPF BONGO B
EECATCHERS       ": FOR f=1 TO 30:
BEEP .05,f: PRINT AT 10,f;"[illegible]";A
T 10,0;a$(1 TO f),: BEEP .05,-f:
 NEXT f
7210 PRINT #1;"           © by Steph
an Schulz": PAUSE 0
7500 REM  ANLEITUNG
7510 POKE 23692,255: PRINT AT 21
,0: PRINT ">>>>>>>>>>> BEECATCHER
 <<<<<<<<<<<";: PRINT : PRINT : P
RINT "Die dicken,fetten Honigbie
nen    ([illegible]) von Bongo Beecatcher
sind   ausgebrochen.Bongo muss si
e      wieder einfangen.Du sollst
```

ZX Spectrum

ZX Spectrum

```
 ihm  dabei helfen.Aber Vorsicht
 !!  Die gefaehrlichen Hornisse
n ([illegible])versuchen,ihn daran zu hin
dern."
7520 PRINT "Bongo kann vier Stic
he ver-     tragen,beim fuenften
 muss er    sterben.Nach jeweils
 fuenf ge-  fangenen Bienen werd
en die rest-lichen vorsichtiger
und fliegen unregelmaessiger.Aus
serdem er-  scheinen immer mehr
Hornissen."
7530 PRINT : PRINT : PRINT "Steu
erung:   rechts.......>X<
7530>PRINT : PRINT : PRINT "Steu
erung:   rechts.......>X<
         Links.........>Z<
         Sprung...>SPACE<": PRINT
 "Viel Glueck....."
7540 IF INKEY$<>"" THEN GO TO 75
40
7550 PAUSE 0: GO TO 160
```

ZX-81

Tonprogramm



Tonprogramm

Dieses Programm ermöglicht es, mit dem sonst tonlosen ZX-81 die verschiedensten Töne bzw. Tonfolgen wiederzugeben. Benötigt wird hierzu ein Mittelwellen-Radio, das dicht neben den ZX-81 gestellt wird.

Das erste Byte der REM-Zeilen dient der Tonbestimmung und kann von 0/hoch bis 255/Tief in 16514 eingepokt werden. Die Länge des jeweiligen Tones wird in 16516 und 16519 festgelegt, wobei in beiden Speicherstellen Werte von 1–255 eingepokt werden können. Sinnvoll ist es jedoch, in 16516 Werte von 1–5 zu poken, da der Ton sonst zu lang werden kann. Hierbei ist zu beachten, daß die hohen Töne kürzer ausfallen wie tiefe Töne. Bei der Programmierung einer Melodie sollte man den tiefen Ton kurz und im Gegensatz hierzu den hohen Ton lang wählen, indem man ihn beispielsweise 2mal programmiert. Mit der Speicherstelle 16535 kann die Tonlage um ganze Blöcke verschoben werden, wobei der sinnvolle Bereich hierbei zwischen 0 und 20 liegen wird. Haben Sie erst einmal eine Melodie programmiert und wollen nun Tonlage und Tonlänge ändern, so ist dies durch Unterbrechen des Programms mit M und neuer Eingabe von L und K möglich.

Pokt man nach der letzten Eingabe eine 0, so geht der Rechner automatisch in das Basic-Programm zurück.

Und so wird's gemacht:

Geben Sie das folgende kleine Basic-Programm, das zur Eingabe des Maschinencodes dient ein und starten Sie es mit RUN. Nun kann der Maschinencode (16516–16580) eingegeben werden.

Hierbei werden natürlich lediglich die Code-Zahlen (60, 6,1 . . . 64) eingegeben. Nach Beendigung dieser Eingabe werden die Zeilen 20–80 des Vorprogramms gelöscht und das Hauptprogramm eingegeben. Starten Sie das Programm nun mit RUN und – sofern Sie es richtig eingegeben haben – viel Spaß!

ZX-81

```
1 REM (70 ZEICHEN EINTIPPEN)
20 POKE 16510,0
30 FOR N=16514 TO 16580
40 INPUT A
50 POKE N,A
60 SCROLL
70 PRINT N,A
80 NEXT N
```

```
16514 - 60     16547 - 16
16515 - 6      16548 - 227
16516 - 1      16549 - 193
16517 - 197    16550 - 16
16518 - 6      16551 - 221
16519 - 200    16552 - 62
16520 - 197    16553 - 127
16521 - 58     16554 - 38
16522 - 130    16555 - 4
16523 - 64     16556 - 219
16524 - 71     16557 - 254
16525 - 62     16558 - 164
16526 - 254    16559 - 200
16527 - 211    16560 - 62
16528 - 255    16561 - 0
16529 - 16     16562 - 58
16530 - 252    16563 - 42
16531 - 175    16564 - 120
16532 - 211    16565 - 254
16533 - 255    16566 - 0
16534 - 6      16567 - 200
16535 - 5      16568 - 50
16536 - 197    16569 - 130
16537 - 58     16570 - 64
16538 - 130    16571 - 42
16539 - 64     16572 - 179
16540 - 71     16573 - 64
16541 - 16     16574 - 35
16542 - 254    16575 - 34
16543 - 193    16576 - 179
16544 - 16     16577 - 64
16545 - 246    16578 - 195
16546 - 193    16579 - 131
               16580 - 64
```

```
   5 PRINT "TONEINGABE"
   7 PRINT "BEENDEN DER EINGABE MIT 300"
  10 FOR N=30721 TO 32766
  20 INPUT A
  30 IF A=300 THEN GOTO 200
  40 POKE N,A
  50 SCROLL
  60 PRINT A
  70 NEXT N
 200 FAST
 210 PRINT "UNTERBRECHEN MIT TASTE M"
 230 PAUSE 100
 232 POKE 16563,1
 234 POKE 16564,120
 236 CLS
 240 RAND USR 16515
 245 PRINT "STOPPEN DES TONS MIT TASTE M"
 250 PRINT "TONLAENGE VERAENDERN DANN L"
 260 PRINT "TONLAGE VERAENDERN DANN K"
 270 PRINT "NOCHMAL HOEREN DANN R"
 280 PRINT "NOCHMAL PROGRAMMIEREN DANN P"
 290 PAUSE 33000
 300 IF INKEY$="P" THEN GOTO 5
 310 IF INKEY$="R" THEN GOTO 232
 320 IF INKEY$="K" THEN GOTO 400
 330 IF INKEY$="L" THEN GOTO 500
 340 CLS
 350 LIST
 400 PRINT "TONLAGE:";PEEK 16535
 410 INPUT A
 420 POKE 16535,A
 430 GOTO 232
 500 PRINT "TONLAENGE:";PEEK 16519
 510 INPUT A
 520 POKE 16519,A
 530 GOTO 232
1000 REM   DEMONSTRATIONTEIL
1001 REM   PROGRAMMIERUNG
1010 LET A=30721
1050 FOR N=20 TO 255
1060 POKE A,N
1070 LET A=A+1
1080 IF A=32000 THEN GOTO 232
1090 NEXT N
1100 GOTO 1030
1200 REM SAVE
1202 FOR N=0 TO 50
1203 SLOW
1204 FAST
1205 NEXT N
1210 SAVE "TOn"
1215 PAUSE 20
1220 GOTO 1000
9900 REM COPYRIGHT BY MARKUS DIEDRICH ROLLSHAUSEN 5/84
```

Aldebaran

Aldebaran
Sie sind Kommandant eines kleinen Raumkreuzers, der sehr gut bewaffnet ist (im unteren Teil des Bildschirms). Sie sollen zu einem Erkundungsflug hinter den feindlichen Linien starten. Ihnen kommen jedoch feindliche, unbewaffnete Raumschiffe entgegen. Im unteren Teil des Bildschirms läuft der Countdown. Ist er bei 0 angelangt, kommen Ihnen weitere, größere Raumschiffe entgegen. Diesen müssen Sie ausweichen.
Das erste Raumschiff ist ein „Zeichen" breit, das letzte Raumschiff ist fünf „Zeichen" breit. Sollten dann Ihre sieben Raumschiffe noch nicht zerstört sein, erhöht sich die Geschwindigkeit und die „Drei-Zeichen-Raumschiffe" kommen Ihnen wieder entgegen. Diese Steigerung hält so lange an, bis Ihre sieben vorhandenen Raumschiffe zerstört sind. Nach jedem SCROLL-Schritt wird der Punkte-Stand um eins erhöht. Die mit „V" gekennzeichneten Raumschiffe können von Ihnen abgeschossen werden. Bei jeder Kugel, die Sie abfeuern, wird Ihr Punkt-Stand um zwei verringert, bei jedem Treffer wird er um fünf erhöht. Unten links wird angezeigt, wieviele Raumschiffe Sie noch besitzen. Oben rechts wird der High-Score angezeigt.
Die Tastenbelegung: 1–5 nach links
6–0 nach rechts
A–G Schuß
Hinweis: Das „Vier-Zeichen-Raumschiff" kann bewußt nicht abgeschossen werden. Da nun aber ein Durchkommen sehr schwer ist, kann das Maschinenprogramm umgeändert werden, indem Sie foldendes eingeben:

POKE 16777,59
POKE 16780,59

Originalzustand wieder herstellen:

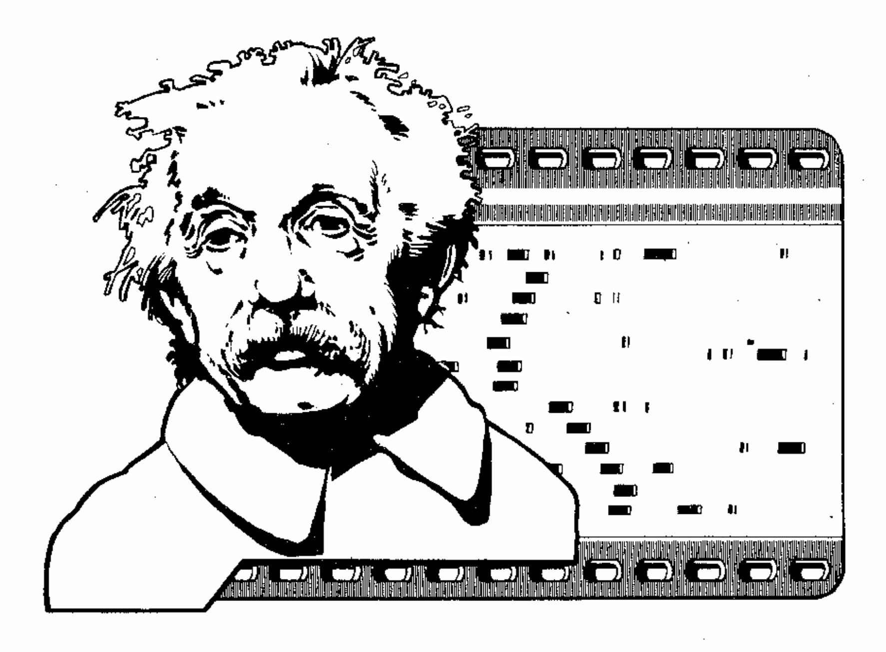
POKE 16777,1
POKE 16780,2

Eingabe des Programms „Aldebaran"
In die erste REM-Zeile müssen 1126 beliebige Zeichen. Dies wird am besten realisiert, wenn man zuerst 704 Zeichen eingibt und dann Newline drückt. Dann holt man die Zeile mit EDIT herunter, fährt mit dem Cursor hinter REM, und gibt dann weitere 422 Zeichen (oder ein paar mehr) ein. Der Rest des Programms ist kein Problem mehr, denn man gibt alles genau nach Listing ein.
Vor dem Starten des Programms mit RUN, sollte man es sicherheitshalber absaven. Dann wird gestartet. Falls das Programm aussteigt, haben Sie irgend etwas zwischen den Zeilen 1050 und 1550 falsch eingegeben. Läuft das Programm dann einwandfrei, können die Zeilen 1000–1550 gelöscht werden (spart Ladezeit).
Dann wird das Programm mit RUN 9990 endgültig abgesavt.
Achtung: Die 1126 beliebige Zeichen in der ersten REM-Zeile lassen sich natürlich auch mit dem in Heft 5 von „Computronic" abgedruckten REM-Schreiber eingeben.

Funktion
In der ersten REM-Zeile wird das 1126 Byte lange Maschinenprogramm abgelegt.
Ab Zeile 1000 werden immer 88 Zahlen in der Stringvariablen Z$ abgelegt, dann wird ins Unterprogramm ab Zeile 1500 gesprungen. Dort werden die Zahlen in die erste REM-Zeile gepokt. Danach kommen die nächsten 88 Zahlen ab Zeile 1120 usw.
In den Zeilen 1800–1820 werden die Variablen G$ und der High-Score H$ sowie die Geschwindigkeit B definiert. In den Zeilen 2100–2180 wird das Bildschirmdisplay aufgebaut.
In den Zeilen 2200–2240 werden verschiedene Daten für das Maschinenprogramm sowie die Geschwindigkeit eingepokt.
In der Zeile 2300 wird in das Maschinenprogramm eingesprungen, das erst wieder verlassen wird, wenn Sie das Maschinenprogramm mit der Break-Taste unterbrechen oder wenn Sie alle sieben Raumschiffe verloren haben.
Nach dem Rücksprung ins BASIC wird der Spielstand in den Zeilen 2400–2450 in der Variablen G$ abgelegt.
G$ wird mit dem High-Score H$ verglichen. Wenn G$ größer als H$ ist (Stringvariablenvergleich), so wird in H$ der neue Wert abgelegt (Zeile 2460).
Ab der Zeile 2490 beginnt das Menue, hier können Sie wählen, ob Sie die Spielanleitung lesen wollen oder ob Sie die Geschwindigkeit des Spiels verändern wollen.
Ab Zeile 2480 beginnt die Geschwindigkeitsänderung; dabei wird die jetzige Geschwindigkeit angezeigt. Danach kann die neue Geschwindigkeit eingegeben werden.
Ab Zeile 3000 beginnt die Spielanleitung.
Ab Zeile 9990 wird das Programm gesaved und anschließend gestartet (Autorun).

```
   1 REM ...1126 BEL. ZEICHEN...
   2 REM
  10 REM ......................
  20 REM ...A.L.D.E.B.A.R.A.N...
  30 REM ......................
  40 REM ..(C)....1984..........
  50 REM .........BY............
  60 REM ...GEORG.HUONKER.......
  70 REM ....ERLENBACHHOF.......
  80 REM .7463..ROS.-LEIDRINGEN.
  90 REM ......................
1000 REM MASCHINENPROGRAMM
1010 FAST
1050 LET Z=16539
1100 LET Z$="033 146 064 034 141
 064 034 143 064 042 012 064 001
 163 002 009 034 132 064 033 146
 064 034 141 064 034 143 064 042
 012 064 001 163 002 009 034 132
 064 033 255 255 034 052 064"
1110 LET Z$=Z$+" 034 147 064 042
 052 064 237 075 147 064 167 237
 066 220 025 068 205 234 067 205
 082 065 205 094 066 205 015 065
 205 239 066 205 006 065 205 049
 065 205 163 065 121 050 139 064
"
1115 GOSUB 1500
1120 LET Z$="205 209 065 205 110
 066 205 239 066 205 096 067 205
 069 066 195 203 064 058 037 064
 006 127 184 192 193 201 042 012
 064 229 001 115 002 017 214 002
 025 229 017 033 000 025 229"
1130 LET Z$=Z$+" 209 225 237 164
 225 001 100 000 009 006 030 035
 054 128 016 251 201 006 025 058
 137 064 128 071 042 130 064 126
 035 034 130 064 144 048 253 128
 042 012 064 001 101 000 009 006
"
1135 GOSUB 1500
1140 LET Z$="000 079 009 195 091
 065 042 134 064 043 124 181 032
 251 201 058 137 064 006 005 164
 202 160 065 006 004 184 202 154
 065 006 003 164 202 145 065 006
 002 164 202 139 065 054 006"
1150 LET Z$=Z$+" 035 054 002 035
 054 059 035 054 001 035 054 134
 201 054 131 035 054 001 035 054
 002 035 054 131 201 054 135 035
 054 059 035 054 004 201 054 006
 035 054 134 201 054 188 201 056
"
1155 GOSUB 1500
1160 LET Z$="139 064 079 058 037
 064 006 247 164 032 002 203 193
 006 239 184 032 002 203 201 005
 253 184 032 002 203 209 006 245
 184 032 004 203 193 203 209 005
 237 184 192 203 201 203 208"
1170 LET Z$=Z$+" 201 042 132 064
 058 139 064 079 203 065 196 236
 065 203 073 196 246 065 203 081
 196 000 066 175 050 139 064 201
 043 126 006 133 184 200 034 132
 064 201 035 126 006 005 184 200
"
1175 GOSUB 1500
1180 LET Z$="034 132 064 201 058
 145 064 006 000 184 192 058 140
 064 203 071 032 026 042 132 064
 034 141 064 205 112 068 205 112
 068 062 002 050 145 064 058 140
 064 203 199 050 140 064 201"
1190 LET Z$=Z$+" 203 079 192 042
 132 064 034 143 064 205 112 068
 205 112 066 062 002 050 145 064
 058 140 064 205 207 050 140 064
 201 042 132 064 126 006 128 184
 229 196 159 068 225 054 166 042
"
1195 GOSUB 1500
1200 LET Z$="141 064 054 149 042
 143 064 054 149 201 042 132 064
 054 126 042 141 064 054 128 042
 143 064 054 128 201 058 145 064
 006 000 184 040 004 061 050 145
 064 058 140 064 079 203 071"
1210 LET Z$=Z$+" 202 161 066 042
 141 064 229 042 012 064 017 196
 000 025 209 167 237 082 210 199
 066 042 141 064 017 033 000 167
 237 082 034 141 064 058 140 064
 079 203 079 200 042 143 064 229
"
1215 GOSUB 1500
1220 LET Z$="042 012 064 017 198
 000 025 209 167 237 082 210 220
 066 042 143 064 017 033 000 167
 237 082 034 143 064 201 175 050
 145 064 058 140 064 203 135 050
 140 064 033 146 064 034 141"
1230 LET Z$=Z$+" 064 195 161 066
 175 050 145 064 058 140 064 203
 143 050 140 064 033 146 064 034
 143 064 201 058 140 064 203 071
 202 020 067 042 141 064 022 000
 020 126 006 059 164 202 056 067
"
1235 GOSUB 1500
1240 LET Z$="006 128 164 194 076
 067 001 033 000 009 062 001 166
 202 252 066 058 140 064 203 079
 200 042 143 064 022 000 020 126
 006 059 184 202 067 067 006 128
 184 194 087 067 001 033 000"
1250 LET Z$=Z$+" 009 062 001 186
 202 031 067 201 058 140 064 203
 223 050 140 064 195 020 067 058
 140 064 203 231 050 140 064 201
 058 140 064 203 239 050 140 064
 195 020 067 058 140 064 203 247
"
1255 GOSUB 1500
1260 LET Z$="050 140 064 201 056
 140 064 203 111 196 188 067 058
 140 064 203 119 196 211 067 058
 140 064 203 095 196 129 067 058
 140 064 203 103 196 139 067 201
 042 141 064 205 149 067 205"
1270 LET Z$=Z$+" 188 067 201 042
 143 064 205 149 067 205 211 067
 201 001 033 000 009 126 006 128
 184 032 006 001 033 000 167 237
 066 062 128 043 043 119 035 119
 035 119 035 119 035 119 006 005
"
1275 GOSUB 1500
1280 LET Z$="197 205 234 067 193
 016 249 201 175 050 145 064 058
 140 064 203 135 203 159 203 175
 050 140 064 033 146 064 034 141
 064 201 175 050 145 064 058 140
 064 203 143 203 167 203 183"
1290 LET Z$=Z$+" 050 140 064 033
 146 064 034 143 064 201 042 012
 064 001 079 000 009 126 006 165
 184 040 003 060 119 201 054 156
 043 126 006 165 184 040 003 060
 119 201 054 156 043 126 006 165
"
1295 GOSUB 1500
1300 LET Z$="184 040 003 060 119
 201 054 156 043 126 060 119 201
 042 147 064 001 050 000 167 237
 066 034 147 064 042 012 064 001
 015 003 009 126 006 156 184 040
 003 061 119 201 054 165 043"
1310 LET Z$=Z$+" 126 006 156 184
 040 003 061 119 201 056 137 064
 006 001 164 229 204 095 068 225
 061 050 137 064 054 161 035 054
 165 033 255 255 034 147 064 034
 052 064 201 062 004 050 137 064
"
1315 GOSUB 1500
1320 LET Z$="042 134 064 001 044
 001 237 066.034 134 064 201 042
 012 064 001 079 000 009 126 006
 156 184 040 003 061 119 201 054
 165 043 126 006 156 164 040 003
 061 119 201 054 165 043 126"
1330 LET Z$=Z$+" 006 156 184 040
 003 061 119 201 054 165 043 126
 061 119 201 042 012 064 001 000
 003 009 006 000 058 136 064 079
 060 050 136 064 167 237 066 126
 006 128 184 040 034 054 128 042
"
1335 GOSUB 1500
1340 LET Z$="012 064 001 133 000
```

```
 009 062 019 054 133 035 061 006
 030 054 128 035 016 251 054 005
 035 054 118 035 006 000 184 032
 234 201 193 193 193 201 000 045
 058 052 051 048 042 055 000"
1345 GOSUB 1500
1350 SLOW
1400 RUN 1800
1500 REM MASCHINENPROGRAMM IN
        DIE ERSTE REM-ZEILE
        POKEN
1510 LET Z=Z+1
1520 POKE Z,VAL Z$( TO 3)
1530 LET Z$=Z$(5 TO )
1540 IF Z$="" THEN RETURN
1550 GOTO 1500
1800 LET G$="00000"
1810 LET H$="00000"
1820 LET B=10
2000 REM SPIEL
2010 CLS
2100 LET A$="
     "
2110 POKE 16418,0
2120 PRINT A$
2130 PRINT "▌          A L D E B A R
 A N                          ▌"
2140 PRINT " SCORE: 00000    HI-S
CORE  ";H$;"▌ "
2150 FOR N=1 TO 20
2160 PRINT A$
2170 NEXT N
2180 PRINT "     AAAAAA           TIME
  59                            "
2200 POKE 16521,5
2210 POKE 16515,0
2220 POKE 16520,0
2230 POKE 16518,255
2240 POKE 16519,6
2300 RAND USR 16540
2400 LET A=PEEK 16396+256*PEEK 1
6397
2410 LET S=A+74
2420 FOR N=1 TO 5
2430 LET S=S+1
2440 LET G$(N)=CHR$ PEEK S
2450 NEXT N
2460 IF G$>H$ THEN LET H$=G$
2470 IF G$=H$ THEN PRINT AT 3,0;
"▌ SIE HABEN SOEBEN DEN REKORD
▌▌ GEBROCHEN
▌"
2480 PRINT AT 5,0;A$;A$
2490 PRINT "▌    1:    SPIEL
             ▌"
2500 PRINT "▌    2:    SPIELANLEITUN
G            ▌"
2510 PRINT "▌    3:    GESCHWINDIGKE
IT AENDERN ▌"
2520 PRINT A$
2530 PRINT "      BITTE TASTE DRUE
CKEN               "
2540 IF INKEY$="1" THEN GOTO 200
0
2550 IF INKEY$="2" THEN GOTO 300
0
2560 IF INKEY$="3" THEN GOTO 258
0
2570 GOTO 2540
2580 REM GESCHW. AENDERN
2590 PRINT A$;A$;A$;"▌JETZIGE G
ESCHWINDIGKEIT: ";B;" ▌"
2600 PRINT A$
2610 PRINT "   BITTE EINGABE GESC
HW. (1-25)         "
2620 POKE 16418,2
2630 INPUT B
2640 IF B<1 OR B>255 THEN GOTO 2
630
2660 GOTO 2000
2900 STOP
3000 REM SPIELANLEITUNG
3010 CLS
3020 PRINT "  A  L  D  E  B  A
R  A  N"
3030 PRINT "  SPIELANLEITUNG:",,
3040 PRINT "SIE SIND KOMMANDANT
EINES KLEI- NEN RAUMKREUZERS, DE
R SEHR GUT  BEWAFFNET IST. SIE S
OLLEN ZU    EINEM ERKUNDUNGSFLUG
 HINTER DEN FEINDLICHEN LINIEN S
TARTEN."
3050 PRINT "IHNEN KOMMEN ABER FE
INDLICHE,   JEDOCH UNBEWAFFNETE
RAUMSCHIFFE ENTGEGEN. IHNEN MUES
SEN SIE AUS-WEICHEN - ODER ABSCH
IESSEN."
3060 PRINT "ES KOENNEN NUR RAUMS
CHIFFE DIE  MIT EINEM ""V"" GEKE
NNZEICHNET      SIND ABGESCHOSSEN
WERDEN.     EINEN CRASH UEBERL
EBEN SIE SECHSMAL."
3070 PRINT "  TASTEN   1-5  :    L
INKS"
3080 PRINT "  TASTEN   6-0  :    R
ECHTS"
3090 PRINT "  TASTEN   A-G  :    S
CHUSS"
3100 PRINT AT 21,31;"▒"
3110 FOR N=1 TO 5
3120 NEXT N
3130 PRINT AT 21,31;" "
3140 FOR N=1 TO 5
3150 NEXT N
3160 IF INKEY$<>"" THEN GOTO 200
0
3170 GOTO 3100
9990 SAVE "ALDEBARAN"
9995 RUN
```



LIBRARY

Eine luxuriöse Diskettenverwaltung für Tippfaule

Zwar gibt es schon eine Menge guter Disketten- oder ganz allgemein Datenverwaltungsprogramme, aber bis 300 Programmnamen mit zugehörigen Daten (Diskvolume/Filetyp/Anzahl der Sektoren etc.) in ein solches Programm eingegeben sind, hat man wunde Finger.

Vorliegendes Programm spart hier die entscheidende Arbeit: Der Diskcatalog der in Drive 0 liegenden Diskette kann per „Einfingerautomatik" programmweise in die Datei übernommen werden; die gedrückte Taste muß dem Anfangsbuchstaben einer der möglichen acht Kategorien entsprechen. Bei (Return) wird das jeweilige Programm nicht übernommen, und sollte man sich einmal getäuscht haben, wird auch angezeigt, daß das Programm schon in der Datei existiert.

Die Eingabefolge (L C C Return) listet z. B. die gesamte Datei nach Kategorien sortiert auf dem Bildschirm; (L A P Return) alphabetisch auf dem Drukker, wobei die möglichen Kommandos jeweils am oberen Bildschirmrand angezeigt werden, ebenso wie der Klartext der bisher eingegebenen Kommandos. Im Editiermodus verbirgt sich außer der schon beschriebenen Automatikfunktion auch die notwendige Möglichkeit zum Suchen und Ändern der Datei sowie natürlich zum Löschen bestimmter Files und eine „Option" zur manuellen Eingabe, damit auch CP/M-Disk ihren Platz in der Library finden. Nach (R Return) bietet das Programm einen Überblick vom Inhalt der Library: Anzahl der Files pro Kategorie, deren Speicherplatz usw. Ferner ist per (D Return) die Möglichkeit gegeben, die wichtigsten DOS-Kommandos (LOCK UNLOCK DELETE) wieder mittels eines Fingerdrucks in bezug auf die in Drive 0 befindlichen Programme zu geben.

Wer jetzt noch die übliche Sortierroutine vermißt, dem sei gesagt, daß man dies auch zwischendurch machen kann. Tatsächlich befindet sich die Datei permanent alphabetisch sortiert auf der Diskette und wird nach jeder Änderung auch dort wieder abgelegt. Dies ist zwar etwas umständlich, hat aber den Vorteil, daß jederzeit (auch über die DOS-Routine) das Programm verlassen werden kann, wenn man es wünscht.

Das Programm ist, sofern es sinnvoll und möglich war, strukturiert geschrieben und hat eine klare Baumstruktur (s. Abb. 1), wobei aus jeder Ebene mittels einfachem Druck der Q-Taste zur Hauptkontrollschleife zurückgesprungen werden kann.

Die Zeilennumerierung würde mit Absicht so vorgenommen, daß jederzeit auf die „Pseudoremarks" verzichtet werden kann (beim Abtippen), sie werden vom Programm nicht angesprungen.

Die wichtigsten Variablen sind in Tabelle 1 angegeben. Sollte es beim Eingeben des Programms Schwierigkeiten geben, steht der Autor gern unter der genannten Anschrift zur Verfügung.

Ein Tip noch: Da beim ersten „RUN" noch keine Datei LIBRARY auf Ihrer Diskette existiert, kommt es logischerweise zum OUT OF DATA-Error in Zeile 48. Hier bitte das Programm mit GOTO 1000 wieder starten und mit (E M E Return) einen beliebigen ersten File editieren – der Fehler tritt dann ab sofort nicht mehr auf.

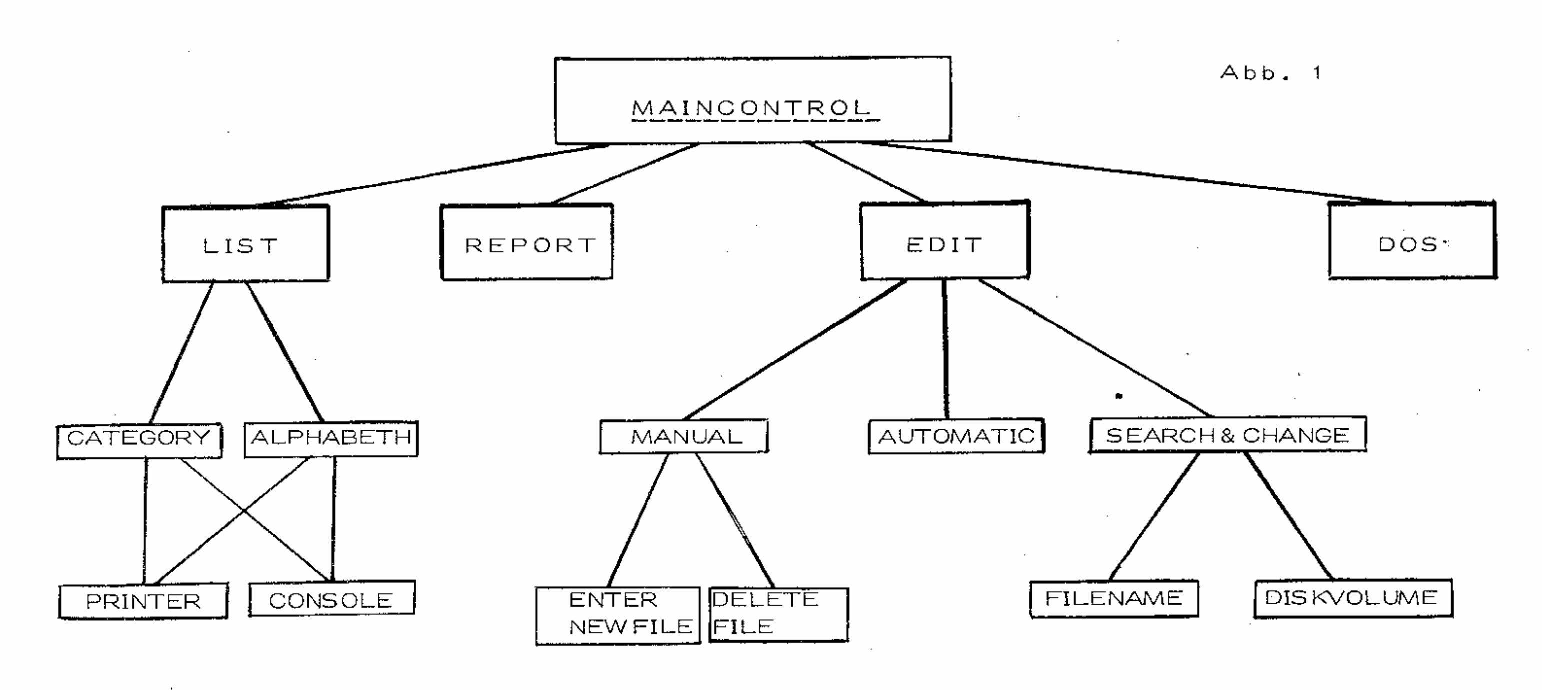


Tabelle 1
Variablenliste

Nummerisch	Zeichenkette	Bedeutung
x,y,z		Schleifenzähler/ Zeiger
i		Zeiger auf Kommandozeile
pf		Printerflag 1 = AN 0 = AUS
cv		Cursorpos Vertikal
ch		Cursorpos Horizontal
n		Anzahl Files
p,pn		Inhalt Cursorpos cv/ch (+ 1)
	C$(X)	Koversationsword
	CL$(X)	Kommandozeile
	K$	Letzte Eingabe
	I$	Informationszeile
	FA$,FD$,FL$,	File-name, -disk-volume, -länge
	FT$,FK$,-(X)	-typ, -kategorie

```
0  GOTO 20
1 "*****************************"
2 "*                           *"
3 "* (C) 1984 BY KLAUS SCHMIDT *"
4 "*                           *"
5 "*      BLESSMANNSTR.1 B     *"
6 "*                           *"
7 "*     2850 BREMERHAVEN 1    *"
8 "*                           *"
9 "*****************************"
10 "*****************************"
11 "* INITIALISION & READ LIB. *"
12 "*****************************"
20  TEXT : HOME : DIM C$(40):D$ =  CHR$ (4)
22 C$ = "      ": FOR X = 0 TO 2:C$ = C$ + C$: NEXT
24 I$ = " READING LIBRARY": GOSUB 120
26  FOR X = 0 TO 40: READ C$(X): NEXT X
28 CL$(0) = C$(0) + C$(1) + C$(2) + C$(3)
30 CL$(1) = C$(4) + C$(5) + C$(10)
32 CL$(2) = C$(11) + "OR " + C$(10)
34 CL$(3) = C$(8) + C$(9) + C$(38) + C$(10)
36 CL$(5) = C$(6) + C$(7) + C$(10)
38 CL$(6) = C$(35) + C$(36) + C$(10)
40 CL$(4) = CL$(2)
42  FOR X = 14 TO 19:CL$(7) = CL$(7) + C$(X): NEXT :CL$(7) = CL$(7) + C$
    (10)
44 CL$(8) = C$(39) + C$(40) + C$(10)
46  PRINT D$;C$(21);C$(20)
48  PRINT D$;C$(23);C$(20): INPUT N: PRINT D$;C$(22);C$(20)
50 N = N + 22
52  DIM FA$(N),FD$(N),FL$(N),FT$(N),FK$(N)
54  IF N = 22 THEN  GOTO 1000
56  PRINT D$;C$(21);C$(20)
58  PRINT D$;C$(23);C$(20): INPUT N
60  FOR X = 0 TO N: INPUT FA$(X),FD$(X),FL$(X),FT$(X),FK$(X): NEXT X
62  PRINT D$;C$(22);C$(20)
64  GOTO 1000
96 "****************************"
97 "* LINE 100-710 SUBROUTINES *"
98 "****************************"
99 " COMMAND/PROMPT - LINEOUT "
100  VTAB 1: HTAB 1: PRINT C$
105  VTAB 1: HTAB 1: INVERSE
110  PRINT CL$(I);: NORMAL : GET K$: PRINT
115  IF K$ = "Q" THEN  POP : POP : GOTO 1000
120  VTAB 2: HTAB 1: PRINT C$
125  VTAB 2: HTAB 1: FLASH
130  PRINT I$
135  NORMAL : RETURN
140  GOSUB 120: GOSUB 100: RETURN
147 "****************************"
148 "* PRINT FILENAME&DATA      *"
149 "****************************"
150  IF L = 0 THEN  INVERSE : PRINT " # VOL SEC T C NAME
      ": NORMAL
```

```
155  IF X < 10 THEN  PRINT " ";
160  PRINT X;" ";FD$(X);" ";FL$(X);" ";FT$(X);" ";FK$(X);" ";
165  IF  LEN (FA$(X)) < 23 THEN  PRINT FA$(X):L = L + 1: RETURN
170 LL =  LEN (FA$(X))
175  FOR ZZ = LL TO 1 STEP  - 1
180  IF  MID$ (FA$(X),ZZ,1) = " " OR  MID$ (FA$(X),ZZ,1) = "-" THEN  PRINT
      LEFT$ (FA$(X),ZZ): HTAB 16: PRINT  RIGHT$ (FA$(X),LL - ZZ):L = L +
     2:ZZ = 1
185  NEXT ZZ: RETURN
197 "***************************"
198 "* SAVE LIBRARY ON DISK    *"
199 "***************************"
200  PRINT D$;C$(21);C$(20): PRINT D$;C$(24);C$(20): PRINT N
205  FOR X = 0 TO N
210  PRINT FA$(X): PRINT FD$(X)
215  PRINT FL$(X): PRINT FT$(X)
220  PRINT FK$(X)
225  NEXT X
230  PRINT D$;C$(22);C$(20): RETURN
247 "***************************"
248 "* FIND CATEGORY           *"
249 "***************************"
250 X = 27
255  IF K$ =  LEFT$ (C$(X),1) THEN  RETURN
260 X = X + 1: IF X > 34 THEN  RETURN
265  GOTO 255
297 "***************************"
298 "*  WAIT AND CLEAR SREEN   *"
299 "***************************"
300  VTAB 23: HTAB 30: PRINT "<RETURN>";: GET X$:L = 0: VTAB 3: HTAB 1
305  FOR Y = 0 TO 20: PRINT C$;: NEXT Y: VTAB 4: RETURN
347 "***************************"
348 "* FIND FILENAME           *"
349 "***************************"
350 X = 0
355  IF FA$(X) = SU$ THEN  RETURN
360  IF X > N THEN  RETURN
365 X = X + 1: GOTO 355
397 "***************************"
398 "* FIND X FOR SORT         *"
399 "***************************"
400 X = 0
405  IF FA$(X) > N$ THEN  RETURN
410  IF X = N + 1 THEN  RETURN
415 X = X + 1: GOTO 405
447 "***************************"
448 "* PUSHDOWN FOR SPACE      *"
449 "***************************"
450  FOR Y = X TO N:Z = Y + 1: GOSUB 550: NEXT Y:N = N - 1: RETURN
487 "***************************"
488 "* PULLUP TO FILL          *"
499 "***************************"
500  FOR Y = N + 1 TO X STEP  - 1:Z = Y - 1: GOSUB 550: NEXT Y:N = N + 1
     : RETURN
547 "***************************"
548 "* SWAP (PULL OR PUSH)     *"
549 "***************************"
550 FL$(Y) = FL$(Z):FD$(Y) = FD$(Z)
555 FT$(Y) = FT$(Z):FK$(Y) = FK$(Z)
560 FA$(Y) = FA$(Z): RETURN
597 "***************************"
598 "* GET SCREENCONTENS CV/CH *"
599 "***************************"
```

```
600 C1 =  INT (CV / 8):C2 = CV - C1 * 8
605 C3 = 1024 + 128 * C2 + 40 * C1 + CH
610 P =  PEEK (C3) - 128:PN =  PEEK (C3 + 1) - 128
615  RETURN
647 "***************************"
648 "* INVENTORY               *"
649 "***************************"
650 FA$(X) = N$:FD$(X) = V$:FL$(X) = L$
655 FT$(X) = T$:FK$(X) = K$: RETURN
697 "***************************"
698 "* SHOW CATEGORIES         *"
699 "***************************"
700  INVERSE : FOR X = 27 TO 34
705  PRINT C$(X);
710  NEXT X: PRINT : NORMAL : RETURN
997 "***************************"
998 "* MAINCONTROL 1000-1100   *"
999 "***************************"
1000  HOME :I$ = "": GOSUB 120
1010 I = 0: GOSUB 100
1020  FOR X = 0 TO 3
1030  IF K$ =  LEFT$ (C$(X),1) THEN I = X:X = 3: GOTO 1080
1040  NEXT X
1050 I$ = C$(25): GOSUB 120: GOTO 1010
1080  HOME :I$ = C$(I):I = I + 1: GOSUB 140
1090  ON I GOSUB 2000,3000,4000,5000
1100  HOME : GOTO 1000
1997 "**************************"
1998 "* LIST CATEGORYCALLY     *"
1999 "**************************"
2000  IF K$ = "A" THEN  GOTO 2500
2010  IF K$ <  > "C" THEN 9990
2020 I = 5:I$ = I$ + C$(5) + C$(12): GOSUB 140
2030  IF K$ = "P" THEN PF = 1:I$ = I$ + C$(6): GOSUB 120: PRINT D$;"PR#1
 ": GOTO 2060
2040  IF K$ <  > "C" THEN 9990
2050 I$ = I$ + C$(7):I = 2: GOSUB 140
2060  FOR K = 27 TO 34: VTAB 3: HTAB 1:L = 0
2070  INVERSE : PRINT C$(K): NORMAL
2080  FOR X = 0 TO N
2090  IF FK$(X) =  LEFT$ (C$(K),1) THEN  GOSUB 150
2100  IF L = 18 AND PF = 0 THEN  GOSUB 300
2110  NEXT X: GOSUB 300
2120  NEXT K: IF PF = 1 THEN PF = 0: PRINT D$;"PR#0"
2130  PRINT "THAT'S ALL !": GOSUB 300: RETURN
2497 "**************************"
2498 "* LIST ALPHABETHICALLY   *"
2499 "**************************"
2500 I = 5:I$ = I$ + C$(4) + C$(12): GOSUB 140
2510  IF K$ = "P" THEN PF = 1:I$ = I$ + C$(6): GOSUB 120: PRINT D$;"PR#1
 ": GOTO 2540
2520  IF K$ <  > "C" THEN  GOTO 9990
2530 I$ = I$ + C$(7):I = 2: GOSUB 140:L = 0
2540  FOR X = 0 TO N
2550  GOSUB 150: IF L >  = 18 AND PF = 0 THEN  GOSUB 300
2560  NEXT X: IF PF = 1 THEN PF = 0: PRINT D$;"PR#0"
2570  GOTO 2130
2997 "**************************"
2998 "*  REPORT                *"
2999 "**************************"
3000  HOME : INVERSE : PRINT "           R E P O R T             "
3010  PRINT "      ";N + 1;" FILES ARE CONTAINING          "
```

```
3020  NORMAL
3030  PRINT "CATEGORY";: HTAB 15: PRINT "#";: HTAB 25: PRINT "SEC";: HTAB
    32: PRINT "KBYTE": PRINT : PRINT
3040  FOR K = 27 TO 34
3050  PRINT C$(K);
3060  FOR X = 0 TO N
3070  IF FK$(X) <  > LEFT$ (C$(K),1) THEN  GOTO 3100
3080 A = A + 1:SP = SP +  VAL (FL$(X)) * .25
3090 SC = SC +  VAL (FL$(X))
3100  NEXT X
3110  HTAB 15: PRINT A;: HTAB 25: PRINT SC;: HTAB 32: PRINT SP
3120 AG = AG + A:SG = SG + SC:ST = ST +.SP
3130 A = 0:SC = 0:SP = 0
3140  NEXT K
3150  PRINT : PRINT "TOTAL";: HTAB 15: PRINT AG;: HTAB 25: PRINT SG;: HTAB
    32: PRINT ST
3160 AG = 0:SG = 0:ST = 0: PRINT : PRINT
3170  GOTO 2130
3997 "***************************"
3998 "* EDIT MANUAL DEL/ENTER   *"
3999 "***************************"
4000  IF K$ = "A" THEN  GOTO 4500
4010  IF K$ = "S" THEN  GOTO 4800
4020  IF K$ <  > "M" THEN  GOTO 9990
4030 I$ = I$ + C$(8):I = 6: GOSUB 140
4040  IF K$ = "E" THEN  GOTO 4140
4050  IF K$ <  > "D" THEN  GOTO 9990
4060 I$ = I$ + C$(36): GOSUB 120
4070  PRINT : INPUT "ENTER FILENAME >";SU$
4080  GOSUB 350: IF X > N THEN  POP :I$ = C$(37): GOTO 9995
4090 L = 0: GOSUB 150:I$ = I$ + FA$(X)
4100 I = 2: GOSUB 140
4110  IF K$ <  >  CHR$ (13) THEN  GOTO 9990
4120 I$ = C$(19) + " " + FA$(X): GOSUB 120
4130  GOSUB 450: GOSUB 200: RETURN
4140 I$ = I$ + C$(35): GOSUB 120
4150  PRINT : INPUT "FILENAME >";N$
4160 SU$ = N$: GOSUB 350: IF X <  = N THEN I$ = "  ALREADY EXISTENT ! ":
    I = 2: GOSUB 140: IF K$ <  >  CHR$ (13) THEN  GOTO 9990
4170  VTAB 5: INPUT "DISK-VOL (XXX)";V$: IF  LEN (V$) <  > 3 THEN  GOTO
    4170
4180  INPUT "# OF SEK (XXX)";L$: IF  LEN (L$) <  > 3 THEN  GOTO 4180
4190  INPUT "FILETYPE (X)  ";T$: IF  LEN (T$) <  > 1 THEN 4190
4200  GOSUB 700
4210  INPUT "CATEGORY      ";K$: GOSUB 250: IF X = 35 THEN  GOTO 4210
4220  IF  LEN (K$) > 1 THEN  GOTO 4210
4230  GOSUB 400: GOSUB 500: GOSUB 650: GOSUB 200: RETURN
4497 "***************************"
4498 "* EDIT AUTOMATIC          *"
4499 "***************************"
4500  PRINT : PRINT "INSERT NEW OR CHANGED DISK"
4510 I$ = I$ + C$(9):I = 2: GOSUB 140
4520  IF K$ <  >  CHR$ (13) THEN  GOTO 9990
4530  HOME : PRINT D$;C$(13):B =  PEEK (37) - 1: IF B > 23 THEN B = 23
4540 V$ = ""
4550 CV = 2: FOR CH = 12 TO 14: GOSUB 600:V$ = V$ +  CHR$ (P): NEXT CH
4560  VTAB 1: GOSUB 700: PRINT "TYPE LETTER TO TAKE OR RETURN TO LET"
4570  FOR CV = 4 TO B:K$ = "":L$ = "":T$ = "":N$ = "":FF = 0
4580  VTAB 4: HTAB 1: PRINT C$
4590  VTAB CV + 1: HTAB 1: PRINT ">";: HTAB 39: GET K$: IF K$ =  CHR$ (1
    3) THEN  GOTO 4700
4600  GOSUB 250: IF X = 35 THEN  GOTO 4590
4610  IF FF = 1 THEN  GOTO 4690
```

```
4620 CH = 1: GOSUB 600:T$ =  CHR$ (P)
4630  FOR CH = 3 TO 5: GOSUB 600:L$ = L$ +  CHR$ (P): NEXT CH
4640  FOR CH = 7 TO 39: GOSUB 600
4650  IF P = 32 AND PN = 32 THEN CH = 39: GOTO 4670
4660 N$ = N$ +  CHR$ (P)
4670  NEXT CH
4680 SU$ = N$: GOSUB 350: IF X <  = N AND FF = 0 THEN  VTAB 4: HTAB 1: PRINT
     "ALREADY EXISTENT! TYPE AGAIN OR RETURN":FF = 1: GOTO 4590
4690  GOSUB 400: GOSUB 500: GOSUB 650
4700  NEXT CV
4710  HOME :I = 2:I$ = "INSERT " + C$(20) + "-DISK ": GOSUB 140
4720  IF K$ <  >  CHR$ (13) THEN  GOTO 4710
4730  HOME : PRINT "SAVING NEW ";C$(20): GOSUB 200: RUN
4797 "**************************"
4798 "* SEARCH AND CHANGE       *"
4799 "**************************"
4800 I = 8:I$ =  LEFT$ (C$(38),7) + " FOR ": GOSUB 140
4805  IF K$ = "D" THEN  GOTO 4900
4810  IF K$ <  > "F" THEN  GOTO 9990
4815 I$ = I$ + C$(40): GOSUB 120
4820  INPUT "FILENAME >";SU$: GOSUB 350
4825  IF X <  = N THEN L = 0: GOSUB 150: GOTO 4835
4830  PRINT C$(37): GOSUB 300:L = 0: GOTO 4800
4835 I$ = "CHANGES ??":I = 2: GOSUB 140
4840  IF K$ <  >  CHR$ (13) THEN  GOTO 9990
4845  VTAB 10
4850  PRINT FA$(X);" TO >";: INPUT N$
4855  PRINT FD$(X);" TO >";: INPUT V$
4860  PRINT FL$(X);" TO >";: INPUT L$
4865  PRINT FT$(X);" TO >";: INPUT T$
4870  PRINT FK$(X);" TO >";: INPUT K$
4875 I$ = "CHANGING": GOSUB 120
4880  GOSUB 450: GOSUB 400: GOSUB 500: GOSUB 650: GOSUB 200: RETURN
4900 I$ =  LEFT$ (I$,12) + " " + C$(39): GOSUB 120:L = 0
4910  INPUT "DISK-VOLUME >";V$: IF  LEN (V$) <  > 3 THEN 4910
4920  FOR X = 0 TO N
4930  IF FD$(X) = V$ THEN  GOSUB 150
4940  IF L = 18 THEN  GOSUB 300:L = 0
4950  NEXT X
4960  GOSUB 300: RETURN
4997 "**************************"
4998 "* EXECUTE DOS             *"
4999 "**************************"
5000  HOME : PRINT D$;C$(13):B =  PEEK (37) - 1:I$ = C$(3)
5010  VTAB 3: HTAB 1: PRINT "TYPE FIRST LETTER OF DOS-COMMAND":I = 7: GOSUB
     140
5020 X = 14:N$ = "":T$ = ""
5030  IF K$ =  LEFT$ (C$(X),1) THEN  GOTO 5060
5040 X = X + 1: IF X > 19 THEN  GOTO 9990
5050  GOTO 5030
5060  FOR CV = 4 TO B:CH = 4: GOSUB 600
5070  POKE C3 - 1, ASC (">"): POKE C3,CV + 125: POKE C3 + 1, ASC ("<")
5080  NEXT CV
5090 I$ = C$(X): GOSUB 120: VTAB 3: HTAB 1: PRINT "TYPE LETTER OF FILE Y
     OU WISH TO ";C$(X)
5100  HTAB 5: GET CV$:CV =  ASC (CV$) - 61
5110  FOR CH = 7 TO 39
5120  GOSUB 600: IF P = 32 AND PN = 32 THEN  GOTO 5140
5130 N$ = N$ +  CHR$ (P)
5140  NEXT CH
5150 CH = 1: GOSUB 600:T$ =  CHR$ (P)
5160  IF X = 14 AND T$ <  > "T" THEN  GOTO 5500
5170  IF X = 15 AND T$ <  > "A" THEN  GOTO 5500
```

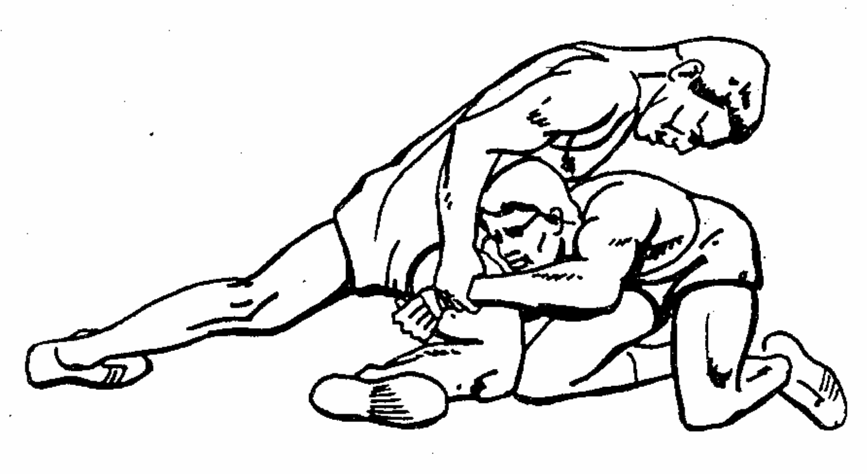
```
5180  IF X = 16 AND T$ <  > "B" THEN  GOTO 5500
5190  IF X = 19 AND  PEEK (C3 - 1) <  > 160 THEN  GOTO 5600
5200 I$ = I$ + N$:I = 2: GOSUB 140
5210  IF K$ <  >  CHR$ (13) THEN  GOTO 9990
5220  PRINT D$;I$: GOTO 5000
5500  HOME :I$ = C$(25): GOSUB 120: INVERSE
5510  VTAB 15: HTAB 5: PRINT N$;" IS ";T$;"-TYP !!"
5530  GOTO 5700
5600  HOME :I$ = C$(25): INVERSE : VTAB 15: HTAB 5: PRINT "FILE IS STILL
      LOCKED !!!"
5700  FOR XZ = 0 TO 5000: NEXT : NORMAL : GOTO 5000
9987 "***************************"
9988 "* FALSE COMMAND           *"
9989 "***************************"
9990  POP :I$ = C$(25)
9995  GOSUB 120: GOTO 1010
9997 "***************************"
9998 "* CONVERSATION-TABLE      *"
9999 "***************************"
10000  DATA     "L'IST ","R'EPORT ","E'DIT ","D'OS "
10100  DATA   "A'LPHABETHICALLY ","C'ATEGORICALLY "
10200  DATA "P'RINTER ","C'ONSOLE ","M'ANUAL ","A'UTOMATIC "
10300  DATA  "Q'UIT ","<RETURN>'FOR ACTION "
10400  DATA     "ON ","CATALOG","EXEC ","RUN ","BRUN ","LOCK ","UNLOCK "
     ,"DELETE ","LIBRARY","OPEN","CLOSE","READ","WRITE"
10500  DATA  "FALSE COMMAND !","SORTING !"
10600  DATA     "U'TILITY ","L'ANGUAGE ","M'ATHEMATIC ","B'USINESS ","G'
    AME ","E'LSE ","D'ATA ","T'EXT "
10700  DATA     "E'NTER NEW FILE ","D'ELETE FILE ","NON EXISTENT !","S'E
    ARCH+CHANGE "
10800  DATA     "D'ISK-VOL ","F'ILENAME+CHANGE "
```

Apple II

Fight

Fight

Fight ist ein Spiel, gedacht für zwei Personen, wo Reaktionsschnelligkeit gefragt ist. Jeder der Spieler zeichnet eine Linie auf dem Bildschirm. Nun muß versucht werden seinen Gegenspieler auszumanövrieren, und zwar so, daß ihm der Platz immer enger bzw. kleiner gemacht wird. Lenkt einer der Spieler gegen irgendeine bereits gezeichnete Linie oder gegen die Wand ist das Spiel verloren. Ein Zeichenspiel also, das bestimmt Spaß bereiten wird. Um das ganze Spiel spannender oder interessanter zu gestalten, kann die Geschwindigkeit zusätzlich gesteigert werden.



```
100  LOMEM: 16400
110 G% = 3 * 256: TEXT : HOME
120  FOR A = G% TO G% + 63
130  READ B: POKE A,B: NEXT
140  DATA 169,2,133,7,162,254,160,2,246,0,181,0,232,56,245,0,202,149,0,202
     ,202,136,208,240,72,24,138,105,4,168,182,0,104
150  DATA 160,0,32,17,244,177,38,37,48,10,240,4,165,7,133,8,32,90,244,162,
     237,198,7,208,204,165,6,32,168,252,96
160 A$ = "    <RETURN> TO PLAY    <ESC> TO QUIT    <SPACE> TO CHANGE SPEED"
170 E% =  - 16384:F% = 128:H% = 8:I% = 0:L% = 83:S% = 255:J% = 253:W% = 23
     8:V% = 236:Y% = 1:Z% = 2:NN(1) = 0:NN(2) = 0
180 K$ = "W":L$ = "S":M$ = "Z":N$ = "X":O$ = ";":P$ = "P":Q$ = ".":R$ = "/
     "
190  VTAB 2: HTAB 11: INVERSE
200  PRINT "###################"
210  HTAB 11: PRINT "                   "
220  HTAB 11: PRINT "   F   I   G   H   T   "
230  HTAB 11: PRINT "                   "
240  HTAB 11: PRINT "###################": NORMAL
250  VTAB 9: PRINT "         COPYRIGHT BY ROBERT SUNA"
260  VTAB 12: PRINT "   PLAYER 1:                PLAYER 2: "
270  VTAB 18: PRINT "UP            W             P"
280  PRINT "DOWN          S             ;"
290  PRINT "LEFT          Z             ."
300  PRINT "RIGHT         X             /"
310  VTAB 16: HTAB 15: PRINT "SPEED (1-5):";: GET T$
320 TT =  ASC (T$): IF TT < 49 OR TT > 53 THEN 310
330  POKE 6,(TT - 49) * 20 + 1: PRINT T$
340  VTAB 23: PRINT : VTAB 14: HTAB 13: PRINT NN(1);: HTAB 36: PRINT NN(2)

350  POKE 8,0: POKE 230,32: CALL 62450
360  HCOLOR= 3
370  HPLOT 0,0 TO 255,0: HPLOT 255,0 TO 255,191
380  HPLOT 255,191 TO 0,191: HPLOT 0,191 TO 0,0
390  VTAB 23: HTAB 1: PRINT  LEFT$ (A$,40);
400  IF  PEEK ( - 16384) > 127 THEN 430
410  FOR II = 1 TO 60: NEXT
420 A$ =  MID$ (A$,2) +  LEFT$ (A$,1): GOTO 390
430  GET B$: IF B$ =  CHR$ (27) THEN 650
440  IF B$ =  CHR$ (32) THEN 310
450  POKE  - 16304,0: POKE  - 16302,0: POKE  - 16297,0
```

Apple II

```
460  POKE S%,Y%: POKE J%,I%: POKE W%,Y%: POKE V%,Z%
470  POKE 254,10: POKE 252,10: POKE 237,245: POKE 235,181
480  IF  PEEK (E%) < F% THEN  CALL G%: IF  PEEK (H%) = I% THEN 480
490  IF  PEEK (H%) > I% THEN 610
500  GET T$: IF  ASC (T$) < L% THEN 560
510  IF T$ = K$ THEN  POKE S%,Y%: POKE J%,Z%: GOTO 480
520  IF T$ = L$ THEN  POKE S%,Y%: POKE J%,I%: GOTO 480
530  IF T$ = M$ THEN  POKE S%,Z%: POKE J%,Y%: GOTO 480
540  IF T$ = N$ THEN  POKE S%,I%: POKE J%,Y%: GOTO 480
550  GOTO 480
560  IF T$ = P$ THEN  POKE W%,Y%: POKE V%,Z%: GOTO 480
570  IF T$ = O$ THEN  POKE W%,Y%: POKE V%,I%: GOTO 480
580  IF T$ = Q$ THEN  POKE W%,Z%: POKE V%,Y%: GOTO 480
590  IF T$ = R$ THEN  POKE W%,I%: POKE V%,Y%: GOTO 480
600  GOTO 480
610  FOR A = 0 TO 200 STEP 20: FOR B = 0 TO A: NEXT : CALL  - 198: NEXT
620  GET T$: TEXT
630 SS =  PEEK (H%):NN(SS) = NN(SS) + 1
640  GOTO 340
650  TEXT : HOME : NEW
```

VC-20

Multigraph

MULTIGRAPH – Programmbeschreibung

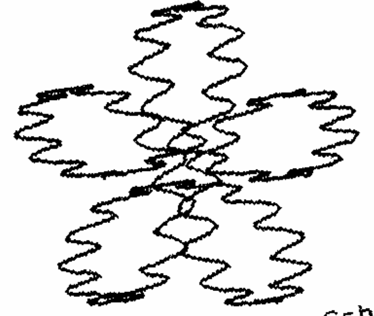
Ein Grafikprogramm für den VC-20 + SUPER EXPANDER VIC 1211A

MULTIGRAPH ist ein Programm, mit dem man auf relativ einfache Weise eine hübsche Grafik erzeugen kann. Je nach Eingabe können Blumen, Sterne, abstrakte Schmetterlinge und viele viele andere phantasievolle Figuren auf dem Bildschrim entstehen.
Und so funktioniert es: Zwei mehr oder weniger modulierte Sinusschwingungen werden aufeinander senkrecht stehend zur Überlagerung gebracht. Dabei werden je zwei benachbarte Punkte miteinander durch eine Gerade verbunden. Läßt man nun nur wenige Punkte berechnen, entstehen Zacken, bei mehr Punkten entstehen Rundungen. Ähnliche Figuren erhält man, wenn man einen Trichter frei pendeln läßt, aus dem ein feiner Sandstrahl auf eine Fläche rieselt. Doch unser Programm macht es uns viel leichter. Nachdem das Programm gestartet wurde, wird durch Drücken der Taste (J) oder (N) kundgetan, ob man selbst Figuren creieren möchte oder das der Rechner tun soll. Wählt man ersteres kann man sich durch Betrachten der Figuren nun entspannen. Im zweiten Fall sind nacheinander einige Werte einzugeben, die man beliebig variieren kann. Allerdings muß folgendes beachtet werden. Die Anzahl der Schwingungen muß eine natürliche Zahl sein, die der Zeichenpunkte ebenfalls. Die Summe der Radien darf den max. Radius nicht überschreiten und die Anzahl der einzelnen Schwingungen (SCHW.) sollte auch eine natürliche Zahl sein. PHI kann alle Werte zwischen -1 und +1 bekommen (andere haben die gleiche Wirkung) und das Frequenzverhältnis (F/F) sollte eine natürliche Zahl zwischen 1 und 10 sein, damit es nicht zu kompliziert wird.
Sind alle Werte eingegeben, wird die Figur gezeichnet und bleibt solange auf dem Bildschirm bis eine Taste gedrückt wird um neue Werte einzugeben. Es empfiehlt sich dabei zunächst einmal die Werte der Beispiele leicht verändert zu verwenden um die Funktion der Werte zu verstehen.
Das Programm ist so geschrieben, daß auch andere Rechner mit Grafikfunktionen leicht verwendet werden können, dabei muß lediglich beachtet werden, daß der Ursprung (0,0) des Koordinatensystems in einer Ecke des Bildschirms liegt und der RM Wert angepaßt wird. Wer die Möglichkeit hat, die Grafik über einen Drucker oder Plotter auszugeben sollte unbedingt davon Gebrauch machen. Auf Papier sieht alles besser aus als eine kurzlebige Bildschirmdarstellung. Dieses Programm verzichtet jedoch auf eine Druckausgabe, weil sie für jeden Drucker/Plotter anders aussehen würde.
Trotzdem viel Vergnügen beim Ausprobieren des Programms!

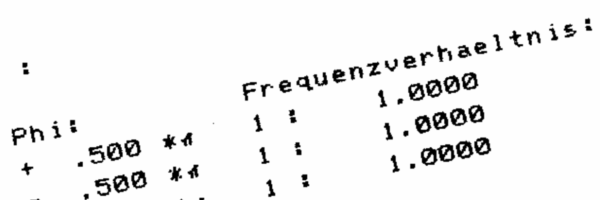
```
1 REM
2 REM   ** MULTIGRAPH **
3 REM
4 REM  SIMULATION DER 2-DIMENSIONALEN
5 REM  UEBERLAGERUNG VON KREISSCHWINGUNGEN
6 REM
7 REM  COPYRIGHT (C) 1984 BY
8 REM  BERND BOLLMANN
9 REM  KIRCHSTR. 1
10 REM 3325 LENGEDE 3
11 REM TEL. (05174)-1069 .
12 REM
13 POKE36879,8:PRINT"■" : REM BILDSCHIRM UND SCHRIFTFARBE
14 PRINT"■\\*O* MULTIGRAPH *O*////BY B.BOLLMANN 1984\\"
15 :
16 :
17 REM VARIABLEN:
18 :
19 REM N = ANZAHL DER UEBERLAGERUNGEN
20 REM PU  = ANZAHL DER ZEICHENPUNKTE
21 REM M   = ZAEHLER FUER N
22 REM T   = ZAEHLER FUER PU
23 REM π   = PI = 3,1415926....
24 RM=511 : REM RECHNERKONSTANTE
25 REM RM = MAXIMALER RADIUS
26 REM RS = RADIENSUMME
27 REM
28 REM SCHWINGUNGSPARAMETER SP(1..N,1..6):
29 REM
30 REM SCHWINGUNGSPARAMETER SP(1..N,1..6):
31 REM SP(M,1) = RADIUS VON KREIS M
32 REM SP(M,2) = SCHWINGUNGEN VON KREIS M/PU
34 REM SP(M,3) = PHASENVERSCHIEBUNG/π VON KREIS M
35 REM SP(M,4) = FREQUENZVERHAELTNIS VON KREIS M
36 REM SP(M,5) = PU PRO SCHWINGUNG VON KREIS M
37 REM SP(M,6) = PHASENVERSCHIEBUNG VON KREIS M
38 :
39 :
40 PRINT"SIMULATION DER         UEBERLAGERUNG VON"
41 PRINT:PRINTCHR$(34)"KREISSCHWINGUNGEN"CHR$(34)
42 PRINT:PRINT"- EINE ERWEITERUNG DER"
43 PRINT"■ LISSAJOUS - FIGUREN .■"
44 PRINT:PRINT"AUTOMATIK EINGABE ?"
45 GETA$:IFA$<>"J"ANDA$<>"N"THEN45
46 IF A$="J" THEN 90
47 GOSUB50:GOSUB70:WAIT198,1:GETA$
48 GRAPHIC0:CLR:RM=511:GOTO47
49 :
50 REM EINGABE DER PARAMETER
51 PRINT"■■EINGABE DER PARAMETER:"
52 INPUT"SCHWINGUNGEN N :";N
53 INPUT"ZEICHENPUNKTE :";PU
54 DIM SP(N,6)
55 FOR M=1 TO N
56 PRINT:PRINT
57 PRINTM". KREISSCHWINGUNG:"
```

```
58 PRINT
59 INPUT"RADIUS  :";SP(M,1)
60 INPUT"SCHW.   :";SP(M,2)
61 INPUT"PHI     :";SP(M,3)
62 INPUT"F/F     :";SP(M,4)
63 SP(M,5)=PU/π/2/SP(M,2)
64 SP(M,6)=SP(M,3)*π
65 NEXT M
66 RETURN
67 :
68 X=0:Y=0
70 REM KURVE ZEICHNEN
71 GRAPHIC2 : REGION 1 :REM GRAPHIK-MODUS EINSCHALTEN
72 REM KOORDINATEN SUMMIEREN:
73 FOR M=1 TO N
74 X=X+SIN(0/SP(M,5))*SP(M,1)+SP(M,1)
75 Y=Y+SIN(SP(M,4)*0/SP(M,5)+SP(M,6))*SP(M,1)+SP(M,1)
76 NEXT M
77 POINT1,X,Y :REM ANFANGSPUNKT SETZEN
78 FOR T=1 TO PU
79 X=0:Y=0 : REM KOORDINATEN LOESCHEN
80 REM KOORDINATEN SUMMIEREN:
81 FOR M=1 TO N
82 X=X+SIN(T/SP(M,5))*SP(M,1)+SP(M,1)
83 Y=Y+SIN(SP(M,4)*T/SP(M,5)+SP(M,6))*SP(M,1)+SP(M,1)
84 NEXT M
85 DRAW1TOX,Y : REM MIT LETZTEM PUNKT VERBINDEN
86 NEXT T
87 RETURN
88 :
89 :
90 REM AUTOMATIK
91 N=INT(RND(1)*5)+1:PU=N*INT(RND(1)*80)+1
92 DIM SP(N,6)
93 FOR M=1 TO N
94 SP(M,1)=INT(RND(1)*(RM-RS)):RS=RS+SP(M,1)
95 SP(M,2)=INT(RND(1)*10)+1
96 SP(M,3)=RND(1)*2-1
97 SP(M,4)=INT(RND(1)*5)+1
98 SP(M,5)=PU/π/2/SP(M,2)
99 SP(M,6)=SP(M,3)*π
100 NEXT M
101 GOSUB70
103 GRAPHIC0:CLR:RM=511:GOTO90
104 :
105 :
130 REM KONFIGURATION:
131 REM VC-20 + SUPER EXPANDER VIC 1211A
132 REM LAENGE 2,5 KB

READY.
```

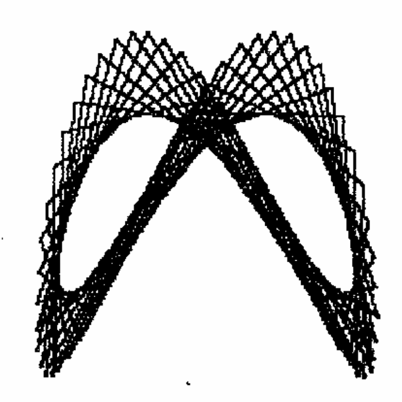


Parameter der Schwingungen 1 bis n :



n:	Radius:	Schwingungen:	Phi:	Frequenzverhaeltnis:
1	250	1	+ .500 **	1 : 1.0000
2	200	4	- .500 **	1 : 1.0000
3	40	50	+ 1.000 **	1 : 1.0000

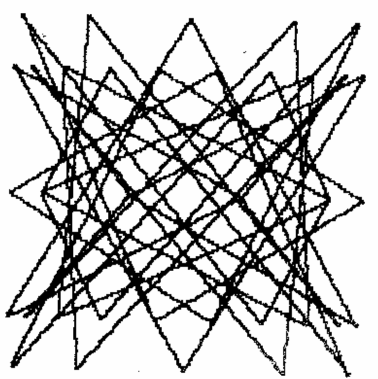
Anzahl der berechneten Punkte : 500



Parameter der Schwingungen 1 bis n :

n:	Radius:	Schwingungen:	Phi:	Frequenzverhaeltnis:
1	487	10	- .757 **	1 : 2.0000

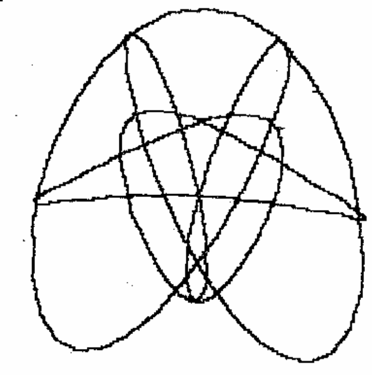
Anzahl der berechneten Punkte : 61



Parameter der Schwingungen 1 bis n :

n:	Radius:	Schwingungen:	Phi:	Frequenzverhaeltnis:
1	500	11	+ .500 **	1 : 5.0000

Anzahl der berechneten Punkte : 40



Parameter der Schwingungen 1 bis n :

n:	Radius:	Schwingungen:	Phi:	Frequenzverhaeltnis:
1	268	10	- .500 **	1 : 1.0000
2	237	6	- .500 **	1 : 2.0000

Anzahl der berechneten Punkte : 250

All Rammer

All Rammer
Ziel des Spiels ist es, 15 Raketensatelliten zu entschärfen. Sie müssen versuchen, Ihr Raumschiff so zu steuern, daß es weder eine Raumstation noch einen der eigenen Container rammt. Für jeden entschärften Satelliten wird ein weiterer Container angehängt. Berührt das Raumschiff die Atmosphäre (Bildschirmrand), so bleibt die Raumkolonne zwar stehen, aber die Zeit läuft weiter, und je länger man braucht, desto niedriger ist der Score. Das Programm besteht aus 2 Teilen, einem Hauptprogramm und einem Vorprogramm, in dem die Zeichen neu definiert werden und die Anleitung enthalten ist. Nach der Eingabe des Vorprogramms „saven" Sie dieses auf Band oder Diskette und geben dann das Hauptprogramm ein, welches dann ebenfalls auf dem jeweiligen Datenträger abgespeichert wird. Gesteuert wird über die Tastatur

»T«,
»F«,
»G«,
»V«. Viel Spaß!

```
3 GOSUB30
5 FORI=7168TO7679:POKEI,PEEK(I+25600):NEXT
10 POKE51,0:POKE52,28:POKE55,0:POKE56,28:CLR:CB=7168
15 READA:IFA=-1THEN195
20 FORN=0TO7:READB:POKECB+A*8+N,B:NEXT
25 GOTO15
30 POKE36879,8
35 PRINT"                 "
40 PRINT"              "
45 PRINT"              "
50 PRINT"              "
55 PRINT"              "
60 PRINT"              "
65 FORI=1TO5
70 PRINT"*RAMMER*"
75 NEXTI
80 PRINT"DRUECKE  'P'"
85 GETA$:IFA$<>"P"THEN85
90 POKE36879,26
95 PRINT"      ALL - RAMMER      ";
100 PRINT" VERSUCHEN SIE MOEG-   LICHST SCHNELL 15     ATOMSATELITTEN ZU "
105 PRINT" ENTSCHAERFEN.RAMMEN   SIE ABER NICHT IHRE   EIGENEN CONTAINER  "
110 PRINT" ODER EINE RAUMSTATION RAMMEN SIE DIE ATMOS- PHARE SO   "
115 PRINT" BLEIBT IHR SCHIFF     STEHEN ES GIBT JE-    DOCH STRAFSEKUNDEN."
125 PRINT"DRUECKE  'P'"
130 GETA$:IFA$<>"P"THEN130
150 POKE36879,25
155 PRINT"":PRINT""
160 PRINT"LENKUNG"
162 PRINT"UEBER DIETASTEN:"
165 PRINT"         TF+GV"
175 PRINT"DRUECKE  'P'"
180 GETA$:IFA$<>"P"THEN180
185 PRINT"BITTE WARTEN"
190 RETURN
195 POKE36869,255
200 POKE36879,25
205 PRINT"A← IHR RAUMSCHIFF"
210 PRINT"B← EIN CONTAINER"
215 PRINT"C← EIN ATOMSATELITT";
220 PRINT"D← EINE RAUMSTATION";
225 PRINT"F← DIE ATMOSPHARE"
230 PRINT"      DRUECKE  'P'"
```

```
235 GETA$:IFA$<>"P"THEN235
250 PRINT"       BITTE WARTEN       TEIL 2 WIRD GELADEN"
275 POKE36869,240:POKE631,131:POKE198,1:END
1000 DATA0,164,85,171,106,60,126,126,60
1005 DATA1,24,36,90,165,165,90,36,24
1010 DATA2,24,126,126,231,231,126,126,24
1015 DATA3,255,129,189,165,189,165,129,255
1020 DATA4,129,60,102,90,90,102,60,129
1025 DATA5,137,68,170,20,181,106,126,255
1030 DATA6,66,165,90,60,60,90,165,66,-1

READY.
```

```
5 DIMS(20):POKE650,128:HI=0:SC=0
6 V1=36878:V2=36875:V3=15:V4=0:V4=36877:V5=36876
8 POKE36869,255
10 POKE36879,8:PRINT"";SPC(220);"        ALL-RAMMER"
11 INPUT"     LEVEL (1-3)";A%
12 IFA%<1ORA%>3THENGOTO10
15 E=0:S(1)=7703:S(2)=7704
17 C1$="FFFFFFFFFFFFFFFFFFFFFF"
18 C2$="F                    F"
20 PRINT
22 PRINT" PRESS 'S' FOR START  "
24 GETA$:IFA$<>"S"THEN24
26 PRINT"";C1$;
28 FORI=1TO19:PRINTC2$;:NEXT:PRINTC1$;
29 FORI=38400TO38905:POKEI,1:NEXT
30 POKES(1),2:POKES(2),1:A=3:B=7
35 FORI=1TO15:GOTO40
36 FORJ=1TO(A%*15)
40 NS=INT(RND(1)*415)+1+7705
50 IFPEEK(NS)<>32THEN40
55 POKENS,A:POKENS+30720,B:NEXT
60 IFA=3THENA=4:B=5:GOTO36
70 Z=0:N=2
100 Z=Z+1:GETA$:IFA$=""THEN107
105 B$=A$:GOTO110
107 FORH=1TO5:NEXT
108 GOSUB900
110 IFB$="V"THENC=+22:GOTO140
111 IFB$="T"THENC=-22:GOTO140
112 IFB$="G"THENC=+1:GOTO140
113 IFB$="F"THENC=-1:GOTO140
120 GOTO100
140 L=S(1):IFPEEK(S(N)+C)=6THEN100
141 GOSUB900
145 IFPEEK(S(N)+C)=4THEN500
150 IFPEEK(S(N)+C)=2THEN300
155 IFPEEK(S(N)+C)=3THENE=1:POKES(N)+C+30720,1:POKEV1,V3:FORI=9TO255:POKEV2,I:NE
XTI
156 POKEV1,0:POKEV2,0:GOTO160
157 IFPEEK(S(N)+C)<>32THEN100
160 FORI=1TON-1
170 S(I)=S(I+1):NEXT
175 S(I)=S(I)+C:POKES(I-1),2:POKES(I),1
180 FORG=1TO50:NEXT
190 POKEL,32
210 IFE<>1THEN100
220 N=N+1:S(0)=L:POKES(1),2
225 IFN>16THEN400
230 FORI=N-1TO0STEP-1
```



```
240 S(I+5-4)=S(I):NEXT:E=0:GOTO100
250 IFNV$="J"THENB$="":GOTO10
300 FORI=NTO1STEP-1
310 POKES(I),0:POKES(I)+30720,2
311 POKEV1,12:FORK=128TO255STEP2:POKEV4,K:NEXTK:POKEV1,0:POKEV4,0
320 FORZ=1TO1000:NEXT:NEXT
330 C$="CONTAINER GERAMMT!":GOTO1000
400 C$="GESCHAFFT!!!!"
450 GOTO1000
500 FORI=NTO1STEP-1
510 POKES(I),5:POKES(I)+30720,2
511 POKEV1,15:FORY=255TO128STEP-2:POKEV2,Y:NEXTY:POKEV1,0:POKEV2,0
520 FORZ=1TO888:NEXT:NEXT
530 C$="RAUMSTATION GERAMMT!":GOTO1000
900 REM
905 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■C:";N-2;"    ■ZEIT:■";Z;
910 RETURN
1000 SC=(N*100-Z)*2:IFSC<=0THENSC=0
1005 PRINT"■■■■■■■■■GAME OVER"
1007 PRINT"■■■■■■";C$;"■"
1010 PRINT"■■■■SCORE:■";SC
1015 IFSC>=HITHENHI=SC
1020 PRINT"■■■■HI-SCORE:■";HI
1030 PRINT:PRINT"■■■■NOCH EINMAL (J/N)■"
1040 GETNV$:IFNV$="N"THEN1070
1050 IFNV$="J"THENB$="":GOTO6
1060 GOTO1040
1070 PRINT"■■■■■■■SCHADE!!■■":END

READY.
```

Atari

Startup

Startup

Ein Hilfsprogramm, programmiert auf den Atari 400

Startup kann jedoch leicht auf die XL-Modelle funktionsfähig gemacht werden. Also ausprobieren, und wenn es nicht klappen sollte, die Redaktion anrufen! (Gilt auch für „Zeilen-Split".)

Dieses Programm bietet alle Programme, die mit „BAS" enden in einem Menne an und lädt bzw. startet seinerseits nach Angabe einer Zahl ein gewünschtes Programm.

Zur Verdeutlichung dieses etwas „verwirrenden" Vorgangs hier ein kurzes Ablaufdiagramm:

Diskette		*Computer*
AUTORUN.SYS	Booten (Urladen)	Autorun. Sys lädt Start
START		
		START bietet Name an
		1. Name 1 2. Name 2 3. Name 3
		Eingabe einer Zahl, z. B. 3
		Prog.-Nr. 3 wird geladen und gestartet
Prog.-Nr. 3		
		Prog. Nr. 3 läuft

Das Programm *AUTORUN.SYS* ist ein Maschinenprogramm und wird von dem Programm AUTORUN.GEN auf Diskette geschrieben, von wo es dann immer beim Einschalten des Computers aktiviert wird. AUTORUN.GEN (Generator) wird also nur einmal aktiviert, um das Maschinenprogramm in ablauffähiger Form auf Diskette zu schreiben.

Mit dem AUTORUN.SYS kann man übrigens auch andere Programme automatisch starten, vorausgesetzt, sie tragen den Namen „START".

```
10 OPEN #1,8,0,"D:AUTORUN.SYS"
20 TRAP 60
30 READ A
40 PUT #1,A
50 GOTO 30
60 END
100 DATA 255,255,106,6,255,6,169,80,141,0,3,169,1,141,1,3,169,63,141,2
110 DATA 3,169,64,141,3,3,169,5,141,6,3,141,5,3,169,0,141,4,3,141
120 DATA 9,3,141,10,3,141,11,3,169,12,141,8,3,32,89,228,16,1,96,162
130 DATA 11,189,0,5,157,0,3,202,16,247,32,89,228,48,6,32,6,5,108,12
140 DATA 0,96,169,193,141,33,3,169,6,141,34,3,96,251,243,51,246,207,6,163
150 DATA 246,51,246,60,246,76,228,243,172,255,6,240,9,185,234,6,206,255,6,160
160 DATA 1,96,140,33,3,169,228,141,34,3,169,155,160,1,96,34,84,82,65,84
170 DATA 83,58,68,34,32,78,85,82,32,32,32,32,32,32,32,20,226,2,227,2
180 DATA 106,6,224,2,225,2,180,6
```

```
1 REM *****************
3 REM STARTUP
4 REM VON RONALD REYER
5 REM 12/83
9 REM *****************
10 DIM A$(20)
20 OPEN #1,6,0,"D:*.*"
30 GRAPHICS 0
40 POKE 703,4
50 SETCOLOR 3,8,13
60 SETCOLOR 0,6,5
70 TRAP 140
80 POKE 752,1
90 INPUT #1,A$
100 A$=A$(3)
110 IF A$(9,11)="BAS" THEN N=N+1:? #6;N;":";A$(1,8):? #6
120 IF N=10 THEN POSITION 20,0:POKE 82,20
130 GOTO 90
140 ? #6;N+1;": DUP.SYS"
150 CLOSE #1
160 POKE 752,0
170 POKE 82,2
180 ? "     WELCHE NUMMER ";
190 INPUT A
200 IF A<1 OR A>N+1 THEN 180
210 IF A=N+1 THEN DOS
220 OPEN #1,6,0,"D:*.*"
230 REM LADEN DES PROGRAMMES
240 FOR J=1 TO A
250 INPUT #1,A$
260 IF A$(11,13)="BAS" THEN NEXT J:GOTO 280
270 GOTO 250
280 A$(1,2)="D:"
290 A$=A$(1,13)
300 A$(11,11)="."
310 A$(12,14)="BAS"
320 IF A$(10-P,10-P)=" " THEN P=P+1:GOTO 320
330 A$(11-P)=A$(11)
340 RUN A$
```


Chopper Fight

Chopper-Fight

für Atari 600/800 XL

Chopper-Fight ist ein schnelles Actionspiel für alle Atari-Fans. Es belegt ca. 11K RAM und wird über Joystick gesteuert.

Die Aufgabe im Spiel besteht darin, eine Basis, aufgebaut von Angreifer aus dem fernen All, zu vernichten. Dafür stehen jedem Spieler drei Versuche bzw. drei Raumschiffe zur Verfügung. Um die Basis zu erreichen, müssen ausgelegte Lichtbarrieren überwunden werden. Mit gezielten

Schüssen können die Barrieren an ihren dicksten Stellen jedoch aufgebrochen und überwunden werden. Schaffen Sie es nicht, den vorgegebenen Weg zu erreichen oder die dickste Stelle einer Barriere ist nicht zu finden, dann drücken Sie den Steuerknüppel nach vorn. Mit einem Hypersprung, gekennzeichnet durch Tonuntermalung und Farbveränderung, können Sie die Lichtfallen doch noch überwinden.
Achten Sie außerdem auf Ihren Energievorrat. Der verringert sich nämlich mit zunehmender Spieldauer und kann das Spiel ebenfalls vorzeitig beenden. Darum haben Sie Gelegenheit, unterwegs auch aufzutanken. Links und rechts der zu durchkreuzenden Bahn sind kleine Tanks mit Energievorrat ausgelegt (F). Wird einer getroffen, hat Ihr Raumschiff vorerst genug Energie, einen Teil des Weges zurückzulegen. Zusätzlich kommen Sie in die Wertung und erhalten pro Treffer 150 Punkte gutgeschrieben. Das Durchschießen einer Lichtbarriere bringt übrigens für jede entstandene Lücke 50 Punkte extra.
Sollten Sie es schaffen, die feindliche Basis zu erreichen, müssen alle vier Reaktoren vernichtet werden. Dafür gibt es dann 5000 Punkte.
Nach dem Erreichen des Zieles, also Vernichtung der Basis und der vier Reaktoren, erhöht sich selbstverständlich auch der Schwierigkeitsgrad. Der macht sich bei Chopper-Fight in der gesteigerten Geschwindigkeit des zu steuernden Raumschiffes bemerkbar.
Folgende Graphikzeichen aus dem Programm möchten wir noch erklären bzw. aufzeigen, wie sie erreicht werden:

☐ = ESC CTRL	☐ = ESC SHIFT CLAER
☐ = ESC SHIFT INSERT	

```
5 REM (c) 1984  CHOPPER-FLIGHT
6 REM BY RICHARD KAEFER
7 REM GOETZENTURMSTRASSE 37
10 POKE 106,PEEK(106)-5:GRAPHICS 0:GOSUB 1710:SETCOLOR 2,0,0:POKE 755,224:SETCOL
OR 0,8,12
20 DL=PEEK(560)+256*PEEK(561):POKE DL+10,7:POKE DL+11,6
30 POSITION 3,5:? "CHOPPER-FLIGHT":POSITION 7,7:? "(c) 1984 by RICHARD KAEFER"
40 ST=(PEEK(106)+1)*256:FOR MOV=0 TO 1023:POKE ST+MOV,PEEK(57344+MOV):NEXT MOV
50 HH=10:SH=3:SC=0:TM=0
60 GRAPHICS 0:GOSUB 1710:SETCOLOR 2,0,0:POKE 755,224
70 POSITION 13,10:? "CHOPPER-FLIGHT":POSITION 12,12:? "by RICHARD KAEFER":POSITI
ON 14,14:? "DRUECKE START"
90 IF PEEK(53279)=6 THEN GOTO 110
100 GOTO 90
110 A=19:B=23:HT=0
120 IF SH<1 THEN GOTO 1680
130 GRAPHICS 0:GOSUB 1710:SETCOLOR 2,0,0:POKE 755,224
140 RESTORE :POKE 82,0
150 FOR MOV=0 TO 7:READ AA:POKE 80+ST+MOV,AA:NEXT MOV
160 DATA 24,24,60,102,219,126,60,24
170 FOR MOV=0 TO 7:READ AA:POKE 96+ST+MOV,AA:NEXT MOV
180 DATA 145,145,145,145,145,145,145,145
190 FOR MOV=0 TO 7:READ AA:POKE 104+ST+MOV,AA:NEXT MOV
200 DATA 60,126,195,207,195,207,207,126
210 FOR MOV=0 TO 7:READ AA:POKE 112+ST+MOV,AA:NEXT MOV
220 DATA 170,0,0,170,0,0,170,0
230 FOR MOV=0 TO 7:READ AA:POKE 120+ST+MOV,AA:NEXT MOV
240 DATA 85,85,85,85,85,85,85,85
250 FOR MOV=0 TO 7:READ AA:POKE 72+ST+MOV,AA:NEXT MOV
260 DATA 231,231,231,231,231,231,231,231
270 FOR MOV=0 TO 7:READ AA:POKE 64+ST+MOV,AA:NEXT MOV
280 DATA 60,126,255,255,255,255,126,60
290 FOR MOV=0 TO 7:READ AA:POKE 672+ST+MOV,AA:NEXT MOV
300 DATA 0,0,60,126,126,60,0,0
310 FOR MOV=0 TO 7:READ AA:POKE 56+ST+MOV,AA:NEXT MOV
320 DATA 67,34,21,89,157,232,111,67
330 FOR MOV=0 TO 7:READ AA:POKE 48+ST+MOV,AA:NEXT MOV
340 DATA 170,170,255,255,255,255,170,170
350 POKE 756,ST/256
360 POSITION 15,10:? "READY!":FOR Z=1 TO 31
370 READ AA:READ BB:FOR ZZ=15 TO 0 STEP -2
375 SOUND 0,AA,10,ZZ:SOUND 1,AA+1,10,ZZ:SOUND 1,BB,10,ZZ:SOUND 1,BB,10,ZZ:NEXT Z
Z
380 DATA 108,0,108,0,53,0,53,0,108,0,108,0,53,0,53,0,121,0,121,0,114,0,114,0,108
,0,108,0,53,0,53,0,108,217,108,0
390 DATA 53,217,53,0,108,217,108,0,53,217,53,0,121,243,121,243,114,144,96,193,10
8,217,108,217,53,108
400 NEXT Z:POSITION 15,10:? "      ":SOUND 0,0,0,0:SOUND 1,0,0,0:SOUND 2,0,0,0
410 POSITION 0,0:? SH;"␣←"
```

```
420 IF SEC<>1 THEN GOTO 460
430 IF SEC=1 THEN COLOR 46:PLOT 0,1:DRAWTO 38,1:COLOR 41
435 PLOT 5,1:PLOT 10,1:PLOT 15,1:PLOT 20,1:PLOT 25,1:PLOT 30,1:PLOT 35,1
440 FOR ZZ=2 TO 10:COLOR 47:PLOT 0,ZZ:DRAWTO 38,ZZ:NEXT ZZ
450 COLOR 38:PLOT 5,4:PLOT 15,4:PLOT 25,4:PLOT 30,4:COLOR 45:PLOT 0,11:DRAWTO 38
,11
460 POSITION 15,0:? "FUEL":COLOR 44:PLOT 19,0:DRAWTO 38,0:II=38
470 DATA 15,24,15,24,15,24,15,24,15,24,15,24,15,24,15,24,15,24,15,24,15,24,15,24
,15,24,15,24,15,24,16,23,17,22
480 DATA 17,22,17,22,17,22,17,22,17,22,17,22,5,33,5,33,5,33,5,33,5,33,5,33,5,33,
5,33,5,33,5,33,5,33,5,33,5,33
490 DATA 10,23,15,23,15,23,15,23,15,23,15,23,15,23,15,23,15,23,15,23,15,23,15,24
,15,24,15,24,15,24,15,24,15,24
500 DATA 16,25,17,26,18,27,19,28,20,29,21,30,22,31,23,32,24,33,25,34,26,35,27,36
,28,37,28,37,28,37,28,37,28,37
510 DATA 28,37,28,37,28,37,28,37,28,37,28,37,5,37,5,37,5,37,5,37,5,37,5,37,5,37,
5,37,5,37,5,37,5,37,5,37,5,37
520 DATA 5,37,5,37,5,37,5,37,5,37,5,37,5,37,5,37,5,30,5,30,5,30,5,30,5,30,5,30,5
,30,5,25,5,25,5,25,5,25,5,25
530 DATA 5,25,5,20,5,20,5,20,5,20,5,15,5,15,5,15,5,15,5,15,5,15,5,15,5,15,5,15,5
,15,5,15,5,15,6,16,7,17,8,18
540 DATA 9,19,10,20,11,21,12,22,13,23,14,24,15,25,16,26,17,27,18,28,19,29,20,30,
21,31,22,32,23,33,24,34,25,35
550 DATA 26,36,27,37,28,38,28,38,28,38,28,38,28,38,28,38,28,38,28,38,28,38,28,38
,28,38,28,38,1,1,1,1,28,38,28,38
560 DATA 28,38,28,38,28,38,28,38,28,38,28,38,28,38,28,38,27,37,5,37,5,37,5,37,5,
37,5,37,15,37,15,37,15,37,15,37
570 DATA 15,37,15,37,15,37,15,37,15,37,15,37,15,37,15,30,15,30,15,30,15,30,15,30
,15,30,15,30,15,24,15,24,15,24
580 DATA 15,24,15,24,15,24,15,24,15,24,15,24,15,24,15,24,15,24,15,24,15,24,15,24
,15,24,15,24,15,24,15,24,15,24
590 DATA 3,24,3,24,3,24,3,24,3,24,3,24,3,24,3,24,3,24,3,24,3,24,3,24,3,24,3,24,3
,24,3,24,3,24,3,24,3,24,3,24,3,24
600 DATA 3,24,3,24,3,24,3,15,3,15,3,15,3,15,3,15,3,15,3,15,3,15,3,15,4,16
605 DATA 5,17,6,18,7,19,8,20,9,21,10,22,11,23,12,24
610 DATA 13,25,14,26,15,24,15,24,15,24,15,24,15,24,15,24,15,24,15,24,15,24,15,24
,15,24,15,24,15,24,15,24,15,24,15
620 DATA 24,15,24,15,24,15,24,15,24,15,24,15,24,15,24,15,24,15,24,15,20,15,20,15
,20,15,17,15,17,15,17,15,17,15,17
630 DATA 15,18,14,19,13,20,12,21,11,22,10,23,9,24,8,25,7,26,6,27,5,28,4,29,3,30,
2,31,1,32,0,33,0,34,0,35,0,36
640 DATA 0,37,0,38,0,38
650 COLOR 42:PLOT A,B
660 IF YY=0 THEN POKE 112+ST,0:POKE 112+ST+6,0
670 IF YY=1 THEN POKE 112+ST,255:POKE 112+ST+6,255
680 YY=YY+1:IF YY>1 THEN YY=0
690 COLOR 32:PLOT II,0:II=II-0.15
700 IF II<19 THEN GOTO 1310
710 IF SEC=1 THEN IF TY=0 THEN POKE 48+ST,85:POKE 48+ST+1,85:POKE 48+ST+6,85:POK
E 48+ST+7,85
720 IF SEC=1 THEN IF TY=1 THEN POKE 48+ST,170:POKE 48+ST+1,170:POKE 48+ST+6,170:
POKE 48+ST+7,170
730 IF SEC=1 THEN TY=TY+1:IF TY>1 THEN TY=0
740 IF SEC=1 THEN TM=TM+1:IF TM>500 THEN TM=0:SEC=0:GOTO 110
750 IF SEC=1 THEN GOTO 820
760 POSITION 19,1:?"[illegible]":LOCATE A,B,C:COLOR 42:PLOT A,B
770 READ AA:READ BB:COLOR 47:PLOT 0,1:DRAWTO AA,1:PLOT BB,1:DRAWTO 38,1
780 IF UU=0 THEN GG=INT(RND(1)*HH):IF GG=0 THEN UU=1:GOSUB 1020
790 IF UU=1 THEN VV=VV+1:IF VV>10 THEN VV=0:UU=0
800 WW=INT(RND(1)*15):IF WW=1 THEN GOSUB 990
810 IF SEC=0 THEN GOTO 830
820 TG=TG+1:IF TG>HH THEN POSITION 0,1:? "":TG=0
830 TRAP 1600
840 SOUND 1,38,0,5
850 IF YY=1 THEN SOUND 0,93,10,8
860 POSITION 5,0:? SC
870 POKE 77,0
880 IF STRIG(0)=0 THEN GOSUB 1050
```

```
890 X=STICK(0):COLOR 32:PLOT A,B
900 IF X=7 OR X=5 THEN A=A+1
910 IF X=11 OR X=9 THEN A=A-1
920 IF SEC=0 THEN IF X=14 OR X=10 OR X=6 THEN GOSUB 1230
930 IF A<2 THEN A=2
940 IF A>38 THEN A=38
950 LOCATE A,B,D:IF C<>42 AND C<>32 AND C<>160 THEN GOTO 1350
960 IF D<>42 AND D<>32 AND D<>160 THEN GOTO 1350
970 SOUND 0,0,0,0
980 GOTO 650
990 XX=INT(RND(1)*BB):IF XX<AA+1 THEN XX=AA+1
1000 COLOR 45:PLOT XX,1
1010 RETURN
1020 COLOR 46:PLOT AA,1:DRAWTO BB,1
1030 EE=INT(RND(1)*BB-1):IF EE<AA+2 THEN EE=AA+2
1040 COLOR 41:PLOT EE-2,1:DRAWTO EE+2,1:RETURN
1050 LOCATE A,B-1,CC:IF CC<>32 AND CC<>160 THEN GOTO 1350
1060 COLOR 20:PLOT A,B-1:SOUND 0,20,0,15:FOR X=1 TO 3:NEXT X
1070 COLOR 40:PLOT A,B-1:SOUND 0,25,0,15:FOR X=1 TO 3:NEXT X
1080 COLOR 20:PLOT A,B-1:SOUND 0,20,0,10:FOR X=1 TO 3:NEXT X:COLOR 32:PLOT A,B-1
1090 FOR TB=B-2 TO 1 STEP -1:LOCATE A,TB,TC
1100 IF TC<>32 AND TC<>160 AND TC<>46 AND TC<>174 THEN GOTO 1130
1110 IF TC=46 OR TC=174 THEN GOTO 1210
1120 NEXT TB:SOUND 0,0,0,0:SOUND 1,0,0,0:RETURN
1130 COLOR 20:PLOT A,TB:SOUND 0,35,0,15:FOR X=1 TO 3:NEXT X
1140 COLOR 40:PLOT A,TB:SOUND 0,40,0,15:FOR X=1 TO 3:NEXT X
1150 COLOR 20:PLOT A,TB:SOUND 0,35,0,10:FOR X=1 TO 3:NEXT X:COLOR 32:PLOT A,TB
1160 IF TC=41 THEN SC=SC+50
1170 IF TC=43 THEN SC=SC+100
1180 IF TC=45 THEN SC=SC+150:GOTO 1440
1190 IF TC=38 THEN SC=SC+500:HT=HT+1:IF HT>3 THEN GOTO 1470
1200 SOUND 0,0,0,0:SOUND 1,0,0,0:RETURN
1210 FOR ZZ=15 TO 0 STEP -2:SOUND 0,121,10,ZZ:SOUND 1,123,10,ZZ:NEXT ZZ:SOUND 0,
0,0,0:SOUND 1,0,0,0
1220 RETURN
1230 SOUND 1,0,0,0:SOUND 2,0,0,0:FOR ZZ=255 TO 0 STEP -10:SOUND 0,ZZ,0,15
1235 SETCOLOR 2,0,PEEK(53770):NEXT ZZ:SETCOLOR 1,0,0:SETCOLOR 2,3,14
1240 TRAP 1580:FOR ZZ=1 TO 10
1250 POSITION 19,1:?"[illegible]":COLOR 42:PLOT A,B:READ AA:READ BB:COLOR 47:PLOT 0,1:DRAW
TO AA,1:PLOT BB,1:DRAWTO 38,1
1260 IF UU=0 THEN GG=INT(RND(1)*HH):IF GG=0 THEN UU=1:GOSUB 1020
1270 COLOR 32:PLOT II,0:II=II-0.5:WW=INT(RND(1)*25):IF WW=1 THEN GOSUB 990
1280 IF II<19 THEN GOTO 1310
1290 NEXT ZZ:FOR ZZ=0 TO 255 STEP 10:SOUND 0,ZZ,0,15
1295 SETCOLOR 2,0,PEEK(53770):NEXT ZZ:SETCOLOR 2,0,0:SETCOLOR 1,0,10
1300 SOUND 0,0,0,0:RETURN
1310 SEC=0:SOUND 1,0,0,0:SOUND 2,0,0,0:FOR ZZ=1 TO 5
1320 POSITION 15,10:? "OUT OF FUEL!":SOUND 0,40,10,15:FOR X=1 TO 60:NEXT X
1330 POSITION 15,10:? "            ":SOUND 0,0,0,0:FOR X=1 TO 60:NEXT X
1340 NEXT ZZ:GOTO 1350
1350 SETCOLOR 1,0,10:SOUND 1,0,0,0:SOUND 2,0,0,0:FOR ZZ=15 TO 5 STEP -0.25
1355 SOUND 0,20,0,ZZ:SETCOLOR 2,0,PEEK(53770):NEXT ZZ
1360 SEC=0:SETCOLOR 2,0,0:COLOR 39:PLOT A,B:SOUND 0,20,0,15:FOR X=1 TO 10:NEXT X
1370 COLOR 39:PLOT A-1,B-1:PLOT A+1,B-1:PLOT A-1,B:PLOT A+1,B:SOUND 0,25,0,15:FO
R X=1 TO 10:NEXT X
1380 COLOR 39:PLOT A-2,B-2:PLOT A+2,B-2:PLOT A-2,B:PLOT A+2,B:SOUND 0,20,0,15:FO
R X=1 TO 10:NEXT X
1390 COLOR 39:PLOT A-3,B-3:PLOT A+3,B-3:PLOT A-3,B:PLOT A+3,B:SOUND 0,20,0,15:FO
R X=1 TO 10:NEXT X
1400 COLOR 39:PLOT A-4,B-4:PLOT A+4,B-4:PLOT A-4,B:PLOT A+4,B:SOUND 0,20,0,15:FO
R X=1 TO 10:NEXT X
1410 SOUND 0,0,0,0:SOUND 1,0,0,0:FOR X=1 TO 200:NEXT X
1420 FOR ZZ=1 TO 25:POSITION 0,1:? "":NEXT ZZ
1430 SH=SH-1:GOTO 110
1440 SOUND 1,0,0,0:FOR ZZ=II TO II+5:IF ZZ>38 THEN GOTO 1460
1450 COLOR 44:PLOT ZZ,0:SOUND 0,ZZ,10,10:FOR X=1 TO 2:NEXT X:SOUND 0,0,0,0:NEXT
ZZ
1460 II=ZZ-1:SOUND 0,0,0,0:SOUND 1,0,0,0:RETURN
1470 SOUND 2,0,0,0:FOR ZZ=1 TO 20
```

```
1480 SETCOLOR 2,0,PEEK(53770):FOR XX=0 TO 255 STEP 20:SOUND 0,XX,10,15:SOUND 1,X
X+2,10,15:NEXT XX
1490 NEXT ZZ
1500 FOR QQ=1 TO 5
1510 FOR ZZ=15 TO 0 STEP -0.5:SOUND 0,20,0,ZZ:SETCOLOR 2,0,ZZ:SETCOLOR 1,0,ZZ:NE
XT ZZ
1520 NEXT QQ
1530 ? "▶":SETCOLOR 1,0,10:COLOR 42:PLOT A,B:SOUND 0,0,0,0:SOUND 1,0,0,0
1540 POSITION 2,5:? "BASIS ZERSTOERT!!! BONUS 5000 PUNKTE"
1550 SC=SC+5000:FOR X=1 TO 500:NEXT X
1560 HH=HH-2:IF HH<2 THEN HH=2
1570 SEC=0:GOTO 110
1580 FOR ZZ=0 TO 255 STEP 5:SOUND 0,ZZ,0,15:SETCOLOR 2,0,PEEK(53770):NEXT ZZ:SET
COLOR 2,0,0:SETCOLOR 1,0,10
1590 SOUND 0,0,0,0:GOTO 1600
1600 SOUND 1,0,0,0:SOUND 2,0,0,0
1610 COLOR 42:PLOT A,B:SOUND 0,0,0,0:POSITION 14,10:? "DU ENTKOMMST!"
1620 FOR X=1 TO 200:NEXT X
1630 POSITION 5,10:? "VERSUCHE DIE BASIS ZU ZERSTOEREN!!":FOR X=1 TO 200:NEXT X
1640 POSITION 5,10:? "ZERSTOERE JEDEN EINZELNEN REACTOR"
1650 FOR X=1 TO 200:NEXT X
1660 POSITION 0,10:? "ZERSTOERE DAS MUTTER SCHIFF.VIEL GLUECK!":FOR X=1 TO 500:N
EXT X
1670 SEC=1:GOTO 110
1680 POSITION 15,10:? "GAME OVER":SETCOLOR 2,0,0:SETCOLOR 1,0,10
1685 SOUND 0,0,0,0:SOUND 1,0,0,0:SOUND 2,0,0,0:SOUND 3,0,0,0
1690 IF PEEK(53279)=6 THEN GOTO 50
1700 GOTO 1690
1710 POKE 16,64:POKE 53774,3:RETURN
```

Atari

Zeilen Split

Zeilen-Split
Programmiert auf den Atari 400 (siehe Startup).

Dieses Hilfsprogramm arbeitet wie AUTORUN. GEN und START nur mit Diskettenstation.

Zeilen-Split numeriert ein anderes Programm neu durch, spaltet alle Zeilen in einzelne Anweisungen auf und gleicht alle Sprünge etc. an die neuen Zeilennummern an. Beginn und Schrittweite der neuen Zeilennummern werden eingegeben. Das alte Programm muß mit LIST „D:Name" auf Diskette stehen, das neue Programm steht mit LIST in der Datei „SPLIT.LST".

Beispiel:
altes Programm

```
10 REM QWERTY: ABC
20 ONX GOTO 10,20,30
30 A=1:B=2:C=3
40 IF X=5 THEN A=1:B=2
50 GOTO Y
```

Beginn: 10
Sprung: 5
neues Programm

```
10 REM QWERTY: ABC
15 ONX GOTO 10,15,20
20 A=1
25 B=2
30 C=3
35 IF X=5 THEN A=1:B=2
```

Fehlerhafte Adressierung in Zeile 50/35 (alt/neu)

Das GOTO Y in Zeile 50 wird natürlich ausgeführt. Jedoch kann SPLIT die Variable nicht an die neuen Verhältnisse angleichen. Der Vorteil dieses Programmes liegt darin, daß ein mit SPLIT bearbeitetes Programm wesentlich übersichtlicher und somit leichter lesbar wird.

Zudem bekommt man mit Hilfe von „Zeilen-Split" wieder genügend Platz, um neues einzufügen.

36864 Horizontale Bildposition (normal 12)
36865 Vertikale Bildposition (normal 38)
36866 Spaltenanzahl des Videobildes (normal 150)
36867 Zeilenanzahl des Videobildes (normal 46)
Bit 0 wählt 8x8 oder 16x8 Matrix der Zeichen
Als sinnvolle Anwendung dieses Registers wäre die Erzeugung einer feststehenden Zeile, welche sich nur durch POKE beschreiben läßt, denkbar.
36869 Bit 4–7 dienen zur Bestimmung der Bildschirmadresse.
Bit 0–3 bestimmt die Lage des Zeichensatzes.
Die folgende Tabelle zeigt den Zusammenhang der ersten 4 Bit's und der Position des Zeichensatzes:

Bit's	Adresse	Bemerkung
0000	32768	ROM Zeichensatz Groß/Grafik
0010	34816	ROM Zeichensatz Klein/Groß
1100	4096	Definierbarer Zeichensatz
1101	5120	Definierbarer Zeichensatz
1110	6144	Definierbarer Zeichensatz
1111	7168	Definierbarer Zeichensatz

```
1 REM
2 REM *** ZEILEN-SPLIT ***
3 REM VON ROLAND REYER
4 REM --- 12/83 ---
10 REM FORMALITAETEN
20 DIM A$(128),B$(128)
30 GRAPHICS 0
40 CLOSE #1
50 ?
60 ? " PROGRAMMNAME ";
70 INPUT A$
80 B$="D1:"
90 B$(4)=A$
100 TRAP 40
110 IF A$="" THEN ? "↖":OPEN #1,6,0,"D1:
*.*":FOR J=1 TO 100:INPUT #1,A$:? A$:NEX
T J
120 TRAP 30
130 OPEN #1,6,0,B$
140 INPUT #1,A$
150 GR=VAL(A$(15))
160 DIM ZN(1000),SP(1000)
170 TRAP 200
180 INPUT #1,A$
190 GOTO 170
200 FR=VAL(A$)
210 CLOSE #1
220 IF FR>GR*2 THEN 280
230 ?"↖↓NICHT GENUEGEND SPEICHER VORHAND
EN!!"
240 ?"↓SPRUNG INS DOS !"
250 FOR T=1 TO 1000
260 NEXT T
270 DOS
280 ?"↖↓↓WELCHEN ZEILENABSTAND WUENSCHEN
SIE ?↓"
290 TRAP 280
300 INPUT ZA
310 ? " BEGINN DER ZEILENNUMMERN ";
320 TRAP 310
330 INPUT ZN
331 ? "↖↓↓WUENSCHEN SIE NUR RENUMBER (1)"
:? "  ODER AUCH ZEILENSPLIT (2) ?↓"
332 TRAP 331
333 INPUT SPLIT
334 IF SPLIT<1 OR SPLIT>2 THEN 331
340 REM 1.DURCHGANG
350 OPEN #1,4,0,B$
360 OPEN #2,8,0,"D1:SPLIT.TXT"
370 TRAP 660
380 ZC=ZC+1
390 INPUT #1,A$
400 SOUND 0,60,10,8
410 ZN(ZC)=VAL(A$)
420 A$=A$(LEN(STR$(ZN(ZC)))+2)
430 SOUND 0,0,0,0
440 SP(ZC)=0
450 J=1:IF SPLIT=1 THEN J=LEN(A$):GOTO 5
40
460 IF A$(J,J)<>CHR$(34) THEN 490
470 J=J+1
480 IF A$(J,J)<>CHR$(34) THEN 470
490 TRAP 510
500 IF A$(J,J+3)="REM " OR A$(J,J+2)="IF
 " THEN J=LEN(A$):GOTO 540
510 IF J=LEN(A$) THEN 540
520 IF A$(J,J)=":" THEN J=J-1:GOTO 540
530 GOTO 640
540 ? #2;A$(1,J)
550 SOUND 0,20,10,8
560 GES=GES+1
570 A$=A$(J)
580 SOUND 0,0,0,0
590 TRAP 610
600 IF A$(2,2)=":" THEN A$=A$(3)
610 SP(ZC)=SP(ZC)+1
620 J=0
630 IF LEN(A$)=1 THEN 370
640 IF J<LEN(A$) THEN J=J+1:GOTO 460
650 GOTO 370
660 CLOSE #1
670 CLOSE #2
680 REM 2.DURCHLAUF
690 GRAPHICS 0
700 OPEN #1,4,0,"D1:SPLIT.TXT"
710 OPEN #2,8,0,"D1:SPLIT.LST"
720 FOR Z=ZN TO GES*ZA+ZN STEP ZA
730 TRAP 1330
740 INPUT #1,A$
750 SOUND 0,60,10,8
760 B$=STR$(Z)
770 B$(LEN(B$)+1)=" "
780 B$(LEN(B$)+1)=A$
790 A$=B$
800 SOUND 0,0,0,0
810 J=1
820 IF A$(J,J)<>CHR$(34) THEN 850
830 J=J+1
840 IF A$(J,J)<>CHR$(34) THEN 830
850 TRAP 930
860 IF  NOT A$(J,J+2)="ON " THEN 930
870 TRAP 900
880 IF A$(J,J+5)=" GOTO " THEN J=J+6:GOT
O 1140
890 TRAP 930
900 IF A$(J,J+6)=" GOSUB " THEN J=J+7:GO
TO 1140
910 J=J+1
920 GOTO 880
930 TRAP 950
```

```
940 IF A$(J,J+3)="REM " THEN 1060
950 TRAP 970
960 IF A$(J,J+4)="THEN " THEN J=J+5:X=VA
L(A$(J)):GOTO 1140
970 TRAP 990
980 IF A$(J,J+4)="GOTO " THEN J=J+5:GOTO
 1140
990 TRAP 1010
1000 IF A$(J,J+4)="TRAP " THEN J=J+5:GOT
O 1140
1010 TRAP 1030
1020 IF A$(J,J+5)="GOSUB " THEN J=J+6:GO
TO 1140
1030 TRAP 1050
1040 IF A$(J,J+7)="RESTORE " AND  NOT (A
$(J+8,J+8)=":" OR J+8>LEN(A$)) THEN J=J+
8:GOTO 1140
1050 IF J<LEN(A$) THEN J=J+1:GOTO 820
1060 ? #2;A$
1070 FOR P=1 TO LEN(A$)
1080 SOUND 0,10,10,4
1090 ?"▖";A$(P,P);
1100 SOUND 0,0,0,0
1110 NEXT P
1120 ?
1130 NEXT Z
1140 TRAP 1510
1150 AZ=VAL(A$(J))
1160 ZGES=0
1170 FOR SZ=1 TO GES
1180 IF ZN(SZ)=AZ THEN NZ=ZGES*ZA+ZN:GOT
O 1220
1190 ZGES=ZGES+SP(SZ)
1200 NEXT SZ
1210 GOTO 1510
1220 AL=LEN(STR$(AZ))
1230 B$=A$(1,J-1)
1240 B$(J)=STR$(NZ)
1250 TRAP 1280
1260 X=LEN(B$)+2
1270 B$(X-1)=A$(J+AL)
1280 J=X-1
1290 A$=B$
1300 TRAP 1050
1310 IF A$(J,J)="," THEN J=J+1:GOTO 1140
1320 GOTO 1050
1330 CLOSE #1
1340 CLOSE #2
1350 REM *** ENDE ***
1360 ?
1370 ?
1380 ? ">>FERTIG "
1390 ?
1400 ?
1410 ? "DAS BEARBEITETE PROGRAMM FINDEN
SIE"
1420 ?
1430 ? "MIT  ENTER'D1:SPLIT.LST"
1450 ? "AUF DIESER DISKETTE"
1460 ?
1470 ?
1480 XIO 33,#1,0,0,"D1:SPLIT.TXT"
1490 END
1500 REM FEHLERMELDUNG
1510 ? "█▯UNZULAESSIGE ADRESSIERUNG IN Z
EILE"    ↖◣
1520 ZGES=0
1530 FOR SUCH=1 TO GES
1540 ZGES=ZGES+SP(SUCH)
1550 IF ZGES>(Z-ZN)/ZA THEN 1570
1560 NEXT SUCH
1570 AZ=ZN(SUCH)
1580 ? AZ;"/";Z;" (ALT/NEU)"
1590 ? "DRUECKEN SIE EINE TASTE"
1600 ?"↑↑↑";
1610 POKE 764,255
1620 IF PEEK(764)=255 THEN 1620
1630 ? "↓↓↓";
1640 POKE 764,255
1650 GOTO 1050
```

Dragon 32

Laser Attack

Laser-Attack

Das Programm ist in Pmode 3 geschrieben, arbeitet also mit viel Farbe. Es umfaßt 535 Programmzeilen und nimmt 22 KB-Arbeitsspeicher in Anspruch. Es ist also relativ lang.

Aus tiefhängenden, verhängnisvollen Wolken bewegen sich feindliche Laserstrahlen auf Deine Stadt. Unterstützt werden die Angreifer durch Raketeneinsatz und Meteoriteneinschläge. Deiner Stadt droht die totale Zerstörung.

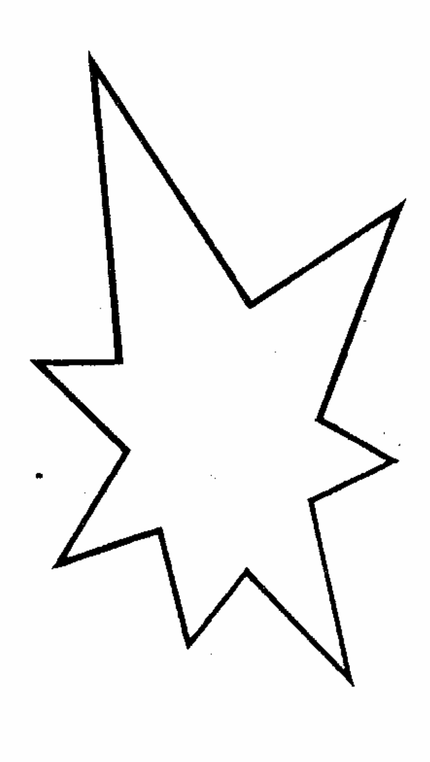
Deine Aufgabe ist es nun, alles Bedrohliche am Himmel abzufangen, bevor die Erde erreicht wird.

Um die feindlichen Laser abzuschießen, muß man natürlich exakt ihre Spitzen treffen. Nur dadurch explodiert der Laser und wird unschädlich gemacht. – Wenn die Laser ungefähr die Mitte des Bildschirms erreicht haben, ändert sich ihre Farbe. Dieses soll nur besagen, daß es langsam Zeit wird, sie abzuschießen, da sie sonst Deine Stadt zerstören. – Wenn Du einmal weniger als 0 Punkte haben solltest (das passiert, wenn Du Deinen Energievorrat aufgebraucht hast), so erscheint keine negative Zahl, sondern eine rote, die jedoch wieder gelb wird, sobald Du über 0 Punkte hast.

Sollte ein Laser in Deiner Stadt auf ein bereits weggesprengtes Stück treffen, so verschwindet er wieder und richtet auch keinen Schaden an; ebenso ist es mit Meteoriten.

Sollten Laser, Meteoriten oder Raketen unterhalb Deiner Laserkanone sein, so kannst Du sie nicht mehr abschießen – Du bist ihnen machtlos ausgeliefert.

Der Schwierigkeitsgrad des Spiels wird mit der Zeit immer höher. Näheres über den Bildschirmaufbau erfährt man in der im Programm enthaltenen Spielanleitung.

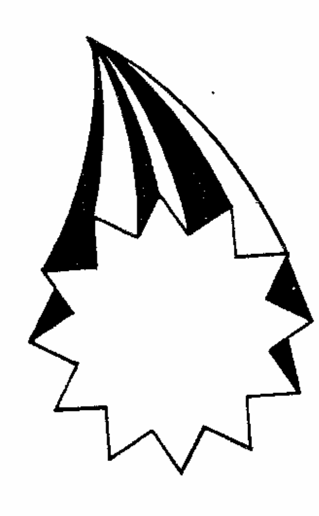


```
10 '===========================
20 '
30 ' ***** LASER - ATTACK *****
40 '
50 '---------------------------
60 '   COPYRIGHTS BY
70 '   KAI COMBUECHEN
80 '   KARLSTR.26,4505 BAD IBURG
90 '   (C) APRIL 1984
100 '
110 '===========================
120 CLS2
130 CLEAR 500
140 PCLEAR 4
150 POKE 65494,0
160 'PROGRAMMKOPF/SPIELNAME'
170 A$=CHR$(128)
180 FOR A=1 TO 122
190 READ B
200 PRINT @B,A$;
210 NEXT A
220 'DATA FUER PROGRAMMKOPF'
230 DATA 33,40,41,42,46,47,48,49,51,52,53,54,55,57,58,59,60
240 DATA 65,71,75,77,83,89,93
250 DATA 97,103,104,105,106,107,110,111,112,115,116,117,121,122,123,124
260 DATA 129,135,139,145,147,153,157
270 DATA 161,162,163,164,165,167,171,173,174,175,176,179,180,181,182,183,185,189
280 DATA 323,324,327,328,329,331,332,333,336,337,341,342,345,348
290 DATA 354,357,360,364,367,370,372,375,377,379
300 DATA 386,387,388,389,392,396,399,400,401,402,404,409,410
310 DATA 418,421,424,428,431,434,436,439,441,443
320 DATA 450,453,456,460,463,466,469,470,473,476
330 'TEXT FUER PROGRAMMKOPF'
340 PRINT @224,CHR$(129)
350 PRINT @225,STRING$(5,131)
360 PRINT @230,CHR$(135)
370 PRINT @248,CHR$(139)
380 PRINT @249,STRING$(6,131)
390 PRINT @255,CHR$(130)
400 PRINT @256,CHR$(132)
410 PRINT @257,CHR$(140)
420 PRINT @258,CHR$(141)
430 PRINT @284,CHR$(142)
440 PRINT @285,STRING$(2,140)
450 PRINT @287,CHR$(136);
460 PRINT @231,"RETTE DEINE STADT";
470 PRINT @259,"VOR DEN TOEDLICHEN LASERN";
480 'FLASHING'
490 FOR A=1 TO 15
500 SCREEN0,0
510 PLAY"T150V1501C02C03C04C05C"
520 FOR B=1 TO 200:NEXT B
530 SCREEN0,1
540 PLAY"T150V1505C04C03C02C01C"
550 FOR B=1 TO 200:NEXT B
560 NEXT A
570 FOR B=1 TO 5000:NEXT B
580 CLS 4
590 PRINT @0,STRING$(32,63);
600 FOR A=1 TO 14:PRINT @32*A,CHR$(63);:PRINT @32*A+31,CHR$(63);:NEXT A
610 PRINT @480,STRING$(31,63);:POKE 1535,127
620 PRINT @232,"ANLEITUNG (J/N)";
630 PLAY"T8V1502CCGGFFL2G"
640 A$=INKEY$:IF A$="" THEN 640
650 IF A$="J" THEN SOUND 1,1:GOTO 660 ELSE IF A$="N" THEN PLAY"T8V10L402CED01B02
L2C":GOTO 1150 ELSE 640
660 CLS 0
```

```
670 PRINT @1,STRING$(30,204);
680 PRINT @481,STRING$(30,195);:POKE 1535,193
690 POKE 1024,200
700 POKE 1055,196
710 POKE 1504,194
720 FOR A=1 TO 14
730 PRINT @32*A,CHR$(202);:PRINT @32*A+31,CHR$(197);
740 NEXT A
750 'SPIELANLEITUNG'
760 'UM NICHT ALLE LEERZEICHEN IN DEN PRINT-ANWEISUNGEN AUSZAEHLEN ZU MUESSEN, H
IER EINE HILFESTELLUNG: DAS ZWEITE ANFUEHRUNGSZEICHEN STEHT IMMER 1 POSITION VOR
 DEM ERSTEN, ALSO: "TEXT ..........................."_(EINE POSITION DAVOR)
770 PRINT @38,"RETTE DEINE STADT !!";
780 PRINT @66,STRING$(28,195);
790 SOUND 40,1
800 PRINT @129,"GENAU DAS IST IN DIESEM SPIEL ";:PRINT @161,"DEINE AUFGABE. KAEM
PFE GEGEN  ";
810 PRINT @193,"DIE ANGREIFENDEN FEINDE, DIE  ";:PRINT @225,"MIT ALLEN MITTELN V
ERSUCHEN,  ";:PRINT @257,"DEINE STADT ZU VERNICHTEN ... ";
820 PRINT @289,"SIE SCHIESSEN MIT MEHREREN     ";:PRINT @321,"LASERN UND SETZEN A
USSERDEM   ";:PRINT @353,"RAKETEN UND METEORITEN EIN. ES";:PRINT @385,"IST ALSO 
HOECHSTE ALARMSTUFE  ";:PRINT @417,"GEGEBEN !!!";:PRINT @449,"(DRUECKE EINE TAST
E !)";
830 IF INKEY$="" THEN 830
840 SOUND 60,1
850 PRINT @129,"NATUERLICH KANNST DU DEINE     ";:PRINT @161,"STADT RETTEN, INDEM
 DU DICH   ";:PRINT @193,"ALS MUTIGER KAEMPFER ZEIGST   ";:PRINT @225,"UND VON D
EINEM LASER GEBRAUCH ";:PRINT @257,"MACHST.                       ";
860 PRINT @289,"EINEN ENTSCHEIDENDEN VORTEIL  ";:PRINT @321,"HAST DU, DA DEIN RA
UMSCHIFF   ";:PRINT @353,"UNVERWUNDBAR IST.             ";
870 PRINT @385,STRING$(30,128);:PRINT @417,STRING$(30,128);
880 SCREEN 0,1
890 IF INKEY$="" THEN 890
900 SOUND 80,1
910 PRINT @129,"DAFUER HAST DU DEINE STADT     ";:PRINT @161,"JEDOCH VERLOREN, WE
NN DIE     ";:PRINT @193,"FEINDE 3 ERFOLGREICHE          ";:PRINT @225,"EINSCHLAE
GE VERZEICHNEN         ";:PRINT @257,"KOENNEN.                      ";
920 PRINT @289,"SOLLTE ES DIR GELINGEN, DIE   ";:PRINT @321,"FEINDLICHEN ANGRIFF
F ABZU-    ";:PRINT @353,"WEHREN, WIRST DU VON DEN EIN- ";:PRINT @385,"WOHNERN W
OHL ZUM 'MISTER        ";:PRINT @417,"WELTRAUM' ERNANNT ...         ";
930 IF INKEY$="" THEN 930
940 SOUND 100,1
950 PRINT @129,"BEVOR DU DIR SCHON DIE JUBELN-";:PRINT @161,"DEN MASSEN VORSTELL
ST, SEI DIR";:PRINT @193,"BESSER ERST EINMAL DER BILD-  ";:PRINT @225,"SCHIRMAUF
BAU ERKLAERT :          ";
960 PRINT @257,"OBEN LINKS WERDEN DEINE PUNKTE";:PRINT @289,"ANGEZEIGT, WOBEI ES
 FUER JEDEN";:PRINT @321,"ZERSTOERTEN LASER 100 PUNKTE  ";:PRINT @353,"GIBT, FUE
R JEDEN METEORITEN    ";:PRINT @385,"300 UND FUER JEDE RAKETE 500. ";
970 PRINT @417,STRING$(30,128);
980 SCREEN 0,1
990 IF INKEY$="" THEN 990
1000 SOUND 120,1
1010 PRINT @129,"RECHTS DANEBEN ERKENNST DU,   ";:PRINT @161,"WIEVIELE EINSCHLAE
GE DEINE     ";:PRINT @193,"STADT NOCH AUSHAELT. RECHTS   ";:PRINT @225,"OBEN WER
DEN DIE ENERGIEEINHEI-";:PRINT @257,"TEN DEINES LASERS ANGEZEIGT.  ";
1020 PRINT @289,"DIE NACH EINEM ANGRIFF UEBRIG-";:PRINT @321,"GEBLIEBENEN ENERGI
EEINHEITEN   ";:PRINT @353,"WERDEN DIR ALS PUNKTE GUTGE-  ";:PRINT @385,"SCHRIEBE
N. ALSO SEI SPARSAM!   ";
1030 IF INKEY$="" THEN 1030
1040 SOUND 140,1
1050 PRINT @129,"SOLLTE DIR DER ENERGIEVORRAT  ";:PRINT @161,"EINMAL AUSGEHEN, S
O BEKOMMST   ";:PRINT @193,"DU EINEN NEUEN, DER DICH JE-  ";:PRINT @225,"DOCH 300
0(!!) PUNKTE KOSTET.   ";
1060 PRINT @257,"WENN EINE RAKETE ODER EIN      ";:PRINT @289,"METEORIT AUFTRITT,
 WERDEN DIE ";:PRINT @321,"LASER INAKTIVIERT, DAMIT       ";:PRINT @353,"SCNELLIG
```

```
KEIT INS SPIEL KOMMT. ";
1070 PRINT @385,"WENN DU DEN BILDSCHIRM AUF     ";:PRINT @417,"EINER SEITE VERLAE
SST, SO      ";
1080 SCREEN 0,1
1090 IF INKEY$="" THEN 1090
1100 SOUND 160,1
1110 PRINT @129,"GELANGST DU AUF DIE ANDERE     ";:PRINT @161,"SEITE. DAS ERHOEHT
 DEINE WEN- ";:PRINT @193,"DIGKEIT ERHEBLICH.            ";:PRINT @225,"DAS SPIE
L WIRD MIT DER ZEIT    ";:PRINT @257,"IMMER SCHWIERIGER ... ES KOM- ";
1120 PRINT @289,"MEN MEHR RAKETEN UND METEORI- ";:PRINT @321,"TEN. DIE RAKETEN F
LIEGEN NUR   ";:PRINT @353,"NOCH EINMAL UEBER DEN BILD-    ";:PRINT @385,"SCHIRM U
ND DIE METEORITEN FAL-";:PRINT @417,"LEN SCHNELLER. VIEL GLUECK !!!";
1130 IF INKEY$="" THEN 1130
1140 'ABFRAGE FUER SPIELBEGINN'
1150 CLS8
1160 PRINT @1,STRING$(30,252);
1170 PRINT @0,CHR$(254);:PRINT @31,CHR$(253);
1180 FOR A=1 TO 14
1190 PRINT @32*A,CHR$(250);:PRINT @32*A+31,CHR$(245);
1200 NEXT A
1210 PRINT@480,CHR$(251);
1220 PRINT STRING$(30,243);
1230 POKE 1535,247
1240 PRINT @66,STRING$(28,175);
1250 PRINT @130,STRING$(28,207);
1260 PRINT @194,STRING$(28,143);
1270 PRINT @322,STRING$(28,191);
1280 PRINT @418,STRING$(28,159);
1290 A$=STRING$(40,255)+STRING$(30,249)+"DRUECKE DIE FIRETASTE FUER SPIELBEGINN.
"+STRING$(30,246)+STRING$(10,255)
1300 'LAUFENDES SCHRIFTBAND'
1310 FOR A=1 TO LEN(A$)+1
1320 PLAY"T40V15L4O3C"
1330 PRINT @257,MID$(A$,A,30);
1340 IF PEEK(65280)=126 OR PEEK(65280)=254 THEN 1380 ELSE FOR B=1 TO 80:NEXT B
1350 NEXT A
1360 TIMER=0
1370 IF TIMER>=150 THEN 1290 ELSE 1370
1380 POKE65495,0:'HOEHERE GESCHWINDIGKEIT'
1390 CLS 2
1400 FOR A=0 TO 95:PRINT @A,CHR$(RND(128)+127);:NEXT A
1410 FOR A=416 TO 510:PRINT @A,CHR$(RND(128)+127);:NEXT A:POKE 1535,179
1420 PRINT @192,STRING$(96,128);
1430 PRINT @228,"NUN GEHT'S GLEICH LOS...";
1440 FOR A=1 TO 31:PLAY"T60O5L4V"+STR$(A)+"G":NEXT A
1450 FOR A=31 TO 1 STEP -1:PLAY"T60V"+STR$(A)+"G":NEXT A
1460 'SPIELVORBEREITUNG/ZEICHENINITIALISIERUNG
1470 CLEAR 500
1480 PMODE 3:PCLS 1:COLOR 2,1
1490 'DIMENSIONIERUNG DER FELDER'
1500 DIM A(8),B(10),C(8),D(5),E(96),G(154),B3(8),B4(8),B$(10),D$(10),H$(5),L(10)
,M(10)
1510 'LEERFELDER'
1520 GET(20,95)-(43,105),B3,G
1530 GET(199,54)-(212,75),B4,G
1540 C=120:H=3:P=3:F=0:V=2:Y=60:R=100:D=246:I=0:O=0:ES=4:ZA=0
1550 'DRAW-STRINGS FUER STADT'
1560 B$(1)="BU7R8U4R4D2R2U9R6D7R4D6R8D4"
1570 B$(2)="R5U2R4U16R2D13R5U3R2E3R6D4R5D5"
1580 B$(3)="R4U14H1L5E1U1H1R7U4R1D4R7G1D1F1L5G1D13"
1590 B$(4)="R5U3R8F3R2F4R3U6R4U6R6D5"
1600 B$(5)="R5D3R5U10R2D8R3D4R3U2R3U2R5"
1610 B$(6)="U3R8U1R4D5R4D2R3"
1620 B$(7)="U10R2D6R2F1R1D3R4U20R9D15R7"
1630 B$(8)="D5R6U3R4U1E4R8F3D7R6"
1640 B$(9)="U13R4D12R4D1R4U8R6D3R4D2R4"
1650 B$(10)="U8E3F3D5R4D4R4U3R4D3F1R2"
```

```
1660 LINE(0,191)-(255,191),PSET
1670 DRAW"BM0,185"
1680 'STADT ZEICHNEN'
1690 FOR A=1 TO 10
1700 DRAW B$(A)
1710 NEXT A
1720 PAINT(1,190),3,2
1730 LINE(0,15)-(255,15),PSET
1740 'RAUMSCHIFF'
1750 DRAW"BM50,155U7D1F2R2E2NU5F2R2E2U1D7H1L3D1L4U1L3"
1760 PAINT(52,153),3,2
1770 GET(47,144)-(68,155),A,G
1780 GET(50,144)-(63,155),D,G
1790 LINE(50,144)-(63,155),PRESET,BF
1800 'RAKETE'
1810 DRAW"BM20,100R5D1NL2D4R3E3NL5R7E2NR2L1U1H1ND3L7NL5H3L3D4L3"
1820 PAINT(27,99),4,2
1830 GET(12,95)-(43,105),B,G
1840 PUT(20,95)-(43,105),B3,PSET
1850 'METEROIT'
1860 DRAW"BM200,60R1F2E2R2E1U2L1U1H1L2H1L1G2D1G1D1"
1870 PAINT(203,60),4,2
1880 GET(199,42)-(210,62),C,G
1890 PUT(199,54)-(210,62),B4,PSET
1900 COLOR 4
1910 'WOLKEN ZEICHNEN'
1920 DRAW"BM0,25R3F2R1F1R1F2R2F1R4E2R1E1R2F1R1E1R2E1R1E1R4F1R2F1R2F1R1F1R1F3D1F2
R1F1R2F2R4E1R1E2U1E1R1E1R1E1R1E1R3E1R1U2E1R5F1R2F1R2F1R1F4"
1930 DRAW"R6E2U1E1R2E1R8E1R1E1R3F1R1F5R1F1R2F3R2E1R1E3R1E1R3E1R3E1R2E2U1E2R1E1R3
F1R2F1R10F1R1F1R8F2R8E2R1E1R5E1R2E2R6F1R5F1R1F1R3F1R3E1R4E1R3E3R2E1R1E1R3F1U1"
1940 LINE(0,16)-(255,16),PSET
1950 PAINT(0,22),4,4
1960 GET(0,17)-(255,39),G,G
1970 COLOR 2
1980 'GAME OVER'
1990 DRAW"BM99,80U8E3R6F2D2L3U1L3G1D6F1R2E1U1L2U2R5D5G2L6H2"
2000 DRAW"BM115,83U11E3R5F3D11L3U5L5D5L2BU8BR2R5U2H1L3G1D2"
2010 DRAW"BM130,83U14R4F4E4R4D14L3U10G4H4D10L2"
2020 DRAW"BM150,83U14R11D3L7D3R4D3L4D2R7D3L10"
2030 DRAW"BM98,102U8E3R7F3D8G3L7H3BR5R2E1U6H1L2G1D6"
2040 DRAW"BM122,105H5U9R3D8F2E2U8R4D9G5"
2050 DRAW"BM134,105U14R11D3L7D3R4D3L4D2R7D3L10"
2060 DRAW"BM150,105U14R9F2D3G3F3D3L3U2H3L1D5L2BM154,97R1E1U1H1L1D2"
2070 LINE(90,65)-(169,110),PSET,B
2080 PAINT(94,66),3,2
2090 GET(90,65)-(169,110),E,G
2100 LINE(90,65)-(169,110),PRESET,BF
2110 'LASER-EXPLOSION'
2110 'LASER-EXPLOSION'
2120 A$="NU2NE2NR2NF2ND2NG2NL2NH2"
2130 'ZIFFERN FUER PUNKTEZAEHLER'
2140 D$(0)="BU1U8BU1BR2R3BR1BD1D8BD1BL4R3BR5"
2150 D$(1)="R5L3U10D1L2D1BD8BR8"
2160 D$(2)="R6NU1L6U5R6U5L6D1BD9BR10"
2170 D$(3)="NU1R6U5NL4U5L6D1BD9BR10"
2180 D$(4)="BU5NU5R6NU5D5BR4"
2190 D$(5)="NU1R6U5L6U5R6BD10BR4"
2200 D$(6)="R6U5L6ND4U5R6BD10BR4"
2210 D$(7)="BR2U4BU1BR2U2BU1BR2U2L6BD10BR10"
2220 D$(8)="U10R6D5NL4D5L4BR8"
2230 D$(9)="R6U10L6D5R6BD5BR4"
2240 GOSUB 4240
2250 GOSUB 4370
2260 'PUNKTE-SCHRIFTZEICHEN'
2270 DRAW"BM2,13C3U10NL2R3BR1BD1D3BD1BL4R2BD4BR6NU9BD1BR2R3BR1BU1U9BR4ND10D3R2D1
BR2BD1D2BR2ND3U7BR4"
2280 DRAW"D10U5R3BR1BD1D2BR2BD1D1BU10D1BL2D2BL2BD1D1BU5BR8"
```



```
2290 DRAW"R7L4D10BR7NR7U5NR5U5R7"
2300 'SCHRIFTZEICHEN FUER ENERGIE DEFINIEREN'
2310 H$(1)="U4NR4D2NR2D2R4BR4"
2320 H$(2)="U4D1R2D1BR1D1R1BD1BR1NU4BR4"
2330 H$(3)="U4R3BD1BR1R0BD1BL2R1BD1BR1D1BR4"
2340 H$(4)="U4R4BD2D2NL2BR4"
2350 H$(5)="R5L3U4L2R5BD4BR3"
2360 'ABGRENZUNGEN AM BILDSCHIRMRAND ZEICHNEN'
2370 COLOR 2
2380 LINE(0,0)-(255,0),PSET
2390 LINE(125,0)-(125,15),PSET
2400 LINE(150,0)-(150,15),PSET
2410 'ENERGIE-SCHRIFTZEICHEN ZEICHNEN'
2420 DRAW"BM179,6;XH$(1);XH$(2);XH$(1);XH$(3);XH$(4);XH$(5);XH$(1);"
2430 COLOR 4
2440 LINE(160,9)-(245,13),PSET,BF
2450 'SPIELBEGINN & BILDSCHIRM'
2460 SCREEN 1,0
2470 GOTO 3160
2480 'SUBROUTINE ZUR JOYSTICKABFRAGE'
2490 B=JOYSTK(0)
2500 A=PEEK(65280)
2510 IF B<8 AND C>3 THEN C=C-3 ELSE IF B<8 AND C<=3 THEN C=233:LINE(3,147)-(24,1
59),PRESET,BF
2520 IF B>55 AND C<233 THEN C=C+3 ELSE IF B>55 AND C=>233 THEN C=3:LINE(233,147)
-(254,159),PRESET,BF
2530 PUT(C,147)-(C+21,159),A,PSET
2540 IF A=127 OR A=255 THEN 2730
2550 COLOR 2
2560 'SCHUSS'
2570 PLAY"T255L4V15O5CO4GCO3GCO2GCO1GC"
2580 FOR B=147 TO 55 STEP -4
2590 LINE(C+10,B-5)-(C+10,B-10),PSET:LINE(C+10,B)-(C+10,B-5),PRESET
2600 NEXT B
2610 LINE(C+10,B-2)-(C+10,B-7),PRESET
2620 LINE(D,9)-(D-E,13),PRESET,BF
2630 D=D-E
2640 IF D>160 THEN 2690
2650 F=F-3000:D=246
2660 COLOR 4:LINE(160,9)-(245,13),PSET,BF:COLOR 2
2670 GOSUB 4240
2680 'SCHUSSAUSWERTUNG(TREFFER/KEIN TREFFER)
2690 IF S=3 THEN GOSUB 3880 ELSE 2710
2700 RETURN
2710 ON S GOSUB 2750,3010
2720 IF G<100 THEN COLOR 4 ELSE COLOR 2
2730 RETURN
2740 'SCHUSSAUSWERTUNG:LASER'
2750 A=C+10
2760 IF A/2<>INT(A/2) THEN A=A-1
2770 FOR W=1 TO H
2780 B=INT(L(W)+M(W))
2790 IF B/2<>INT(B/2) THEN B=B-1
2800 IF A=B THEN 2820 ELSE NEXT
2810 RETURN
2820 DRAW"BM"+STR$(INT(L(W)+M(W)))+","+STR$(N+1)
2830 FOR A=1 TO 16
2840 DRAW"S"+STR$(A)+A$
2850 PLAY"T50V"+STR$(A)+"O1F"
2860 NEXT A
2870 FOR A=16 TO 1 STEP -1
2880 DRAW"C1S"+STR$(A)+A$
2890 NEXT A
2900 FOR A=N+1 TO 20 STEP -1
2910 LINE(L(W)+M(W),A)-(L(W),A-1),PRESET
2920 L(W)=L(W)-M(W)
```

```
2930 NEXT A
2940 L(W)=0
2950 IF I=1 THEN RETURN
2960 F=F+100
2970 GOSUB 4240
2980 P=P-1
2990 IF P>0 THEN RETURN ELSE GOTO 3160
3000 'SCHUSSAUSWERTUNG:RAKETE'
3010 IF K+10<C+10 AND C+10<K+31 THEN 3020 ELSE RETURN
3020 PLAY"T255V15L4O1CC#DD#EFF#GG#AA#BO2CC#DD#EFF#GG#AA#BO3CC#DD#EFF#GG#AA#BO4CC
#DD#EFF#GG#AA#BO5CC#DD#EFF#GG#AA#B"
3030 FOR A=1 TO 15
3040 PUT(K,J)-(K+31,J+10),B,PSET
3050 PLAY"T60V15L4O1C"
3060 FOR B=1 TO 20:NEXT B
3070 PUT(K,J)-(K+31,J+10),B3,PSET
3080 PLAY"T60V15L4O1C"
3090 FOR B=1 TO 20:NEXT B
3100 NEXT A
3110 F=F+500
3120 GOSUB 4240
3130 G=N+1
3140 GOTO 3380
3150 'HAUPTROUTINE/LASER'
3160 FOR W=1 TO 10
3170 L(W)=0:M(W)=0
3180 NEXT W
3190 X=MEM
3200 G=20
3210 PUT(0,17)-(255,39),G,PSET
3220 COLOR 4
3230 IF O=0 THEN 3290
3240 LINE(160,9)-(245,13),PSET,BF
3250 F=F+(D-160)*5
3260 D=246
3270 GOSUB 4240
3280 PLAY"T4L4V15O2CO1L8GGL4G#L8GP4L4BO2C"
3290 H=H+1:P=H:S=1:O=1
3300 IF H=11 THEN 4610
3310 FOR W=1 TO H
3320 B=RND(2)
3330 IF B=1 THEN A=1 ELSE A=-1
3340 L(W)=RND(253)+2
3350 M(W)=A*(3*RND(0)+.03)
3360 IF L(W)+170*M(W)>248 OR L(W)+170*M(W)<7 THEN 3340
3370 NEXT W
3380 S=1
3390 FOR N=G TO 188
3400 IF G<100 THEN COLOR 4 ELSE COLOR 2
3410 FOR W=1 TO H
3420 IF L(W)=0 THEN 3450
3430 LINE(L(W),N)-(L(W)+M(W),N+1),PSET
3440 IF G>160 THEN GOSUB 4430
3450 NEXT W
3460 IF G>146 THEN 3480
3470 E=2:GOSUB 2490
3480 FOR W=1 TO H
3490 IF L(W)=0 THEN 3510
3500 L(W)=L(W)+M(W)
3510 NEXT W
3520 A=RND(R)
3530 IF A=10 THEN 3620 ELSE IF A=20 THEN 3980
3540 G=G+1
3550 NEXT N
3560 FOR W=1 TO H
```

```
3570 IF L(W)=0 THEN 3590
3580 I=1:GOSUB 2900:I=0:L(W)=0
3590 NEXT W
3600 GOTO 3160
3610 'METEROIT'
3620 T=RND(220)+10
3630 S=3:G=N+1
3640 FOR U=40 TO 168 STEP V
3650 PUT(T,U)-(T+11,U+21),C,PSET
3660 IF U>133 THEN 3690
3670 E=4:GOSUB 2490
3680 GOSUB 2490
3690 NEXT U
3700 'METEROIT SCHLAEGT IN STADT EIN'
3710 FOR A=0 TO 11
3720 IF PPOINT(T+A,190)=3 THEN 3730 ELSE NEXT A:GOTO 3850
3730 FOR WE=1 TO 10
3740 SCREEN 1,1
3750 PLAY"T250V20L401GG#AA#AG#G"
3760 FOR ZF=1 TO 100:NEXT ZF
3770 SCREEN 1,0
3780 PLAY"T250V20L401F#FED#EFF#"
3790 FOR ZF=1 TO 100:NEXT ZF
3800 NEXT WE
3810 LINE(T-3,160)-(T+14,190),PRESET,BF
3820 GOSUB 4370
3830 GOSUB 5330
3840 GOTO 3860
3850 PUT(T,U)-(T+13,U+18),B4,PSET
3860 GOTO 3380
3870 'METEROIT GETROFFEN ?'
3880 IF T=<C+10 AND C+10=<T+11 THEN 3890 ELSE RETURN
3890 PUT(T,U)-(T+13,U+21),C,PSET
3900 PLAY"T50V15L405EC04G05C04GEGECEC03G04C03GEGECEC02G03C02GEGEC01G02C01GEGEC"
3910 FOR A=1 TO 500:NEXT A
3920 PUT(T,U)-(T+13,U+21),B4,PSET
3930 F=F+300
3940 GOSUB 4240
3950 COLOR 4
3960 GOTO 3380
3970 'RAKETE'
3980 S=2
3990 PLAY"T250L4V2501GF#FEFF#GF#FEFF#GF#FEFF#GF#FEFF#G"
4000 FOR J=60 TO 180 STEP Y
4010 FOR K=5 TO 222 STEP 8
4020 PUT(K,J)-(K+31,J+10),B,PSET
4030 IF J=180 THEN 4090
4040 E=6:GOSUB 2490
4050 NEXT K
4060 PUT(K,J)-(K+23,J+10),B3,PSET
4070 NEXT J
4080 'RAKETE SCHLAEGT IN STADT  EIN'
4090 IF PPOINT(K+32,189)=1 THEN 4050 ELSE 4100
4100 FOR A=1 TO 10
4110 SCREEN 1,1
4120 PLAY"T250V20L402GG#AA#AG#G"
4130 FOR B=1 TO 100:NEXT B
4140 SCREEN 1,0
4150 PLAY"T250V20L402F#FED#EFF#"
4160 FOR B=1 TO 100:NEXT B
4170 NEXT A
4180 LINE(K,160)-(K+36,190),PRESET,BF
4190 GOSUB 4370
4200 GOSUB 5330
4210 G=N+1
4220 GOTO 3380
```

```
4230 'PUNKTEZAEHLUNG'
4240 DRAW"S4C2"
4250 IF F<0 THEN DRAW"C4"
4260 E$=STR$(F)
4270 A=LEN(E$)
4280 LINE(64,3)-(123,13),PRESET,BF
4290 DRAW"BM64,13"
4300 FOR B=A-1 TO 1 STEP -1
4310 ZF=A-B+1
4320 WE=VAL(MID$(E$,ZF,1))
4330 DRAW D$(WE)
4340 NEXT B
4350 RETURN
4360 'ZAEHLER FUER FEINDLICHE EINSCHLAEGE IN DEINER STADT
4370 ES=ES-1
4380 IF ES<0 THEN 4830
4390 LINE(130,2)-(145,14),PRESET,BF
4400 DRAW"BM134,13S4C3"+D$(ES)+"C2"
4410 RETURN
4420 'LASER ERREICHT STADT'
4430 IF PPOINT(L(W)+M(W),N+2)=3 THEN 4440 ELSE RETURN
4440 FOR A=1 TO 10
4450 SCREEN 1,1
4460 PLAY"T250V20L4O3GG#AA#AG#G"
4470 FOR B=1 TO 100:NEXT B
4480 SCREEN 1,0
4490 PLAY"T250V20L4O3F#FED#EFF#"
4500 FOR B=1 TO 100:NEXT B
4510 NEXT A
4520 LINE(L(W)+M(W)-8,160)-(L(W)+M(W)+8,190),PRESET,BF
4530 GOSUB 5330
4540 GOSUB 4370
4550 I=1:GOSUB 2900:I=0
4560 L(W)=0
4570 P=P-1
4580 IF P>0 THEN RETURN ELSE GOTO 3160
4590 RETURN
4600 'ZWISCHENBILANZ'
4610 CLS 7
4620 ZA=ZA+1:H=3:V=V+1:R=R-10:O=0:IF ZA>=? THEN Y=120
4630 IF ZA=4 THEN 4710
4640 PRINT @33,STRING$(30,159);:PRINT @417,STRING$(30,159);
4650 PRINT @106,"GRATULIERE";
4660 PRINT @161,"DER";ZA;:PRINT @166,".ANGRIFF WAERE GLUECKLICH";:PRINT @233,"UE
BERSTANDEN !";
4670 PRINT @290,"ZAHL DER NOCH BEVORSTEHENDEN";:PRINT @362,"ANGRIFFE :";4-ZA;
4680 PRINT @483,"DRUECKE DIE FIRETASTE ....";
4690 IF PEEK(65280)=126 OR PEEK(65280)=254 THEN SCREEN 1,0:GOTO 3160 ELSE 4690
4700 'ALLE FEINDLICHEN ANGRIFFE UEBERSTANDEN'
4710 CLS 4:PRINT @43,"super";:POKE 1072,32:FOR A=1073 TO 1076:POKE A,33:NEXT A
4720 PRINT @97,"EINE UNGLAUBLICHE LEISTUNG !!!";:PRINT @161,"ES IST DIR SOEBEN G
ELUNGEN,    ";
4730 PRINT @193,"DEINE STADT VOR DEN FEIND-     ";:PRINT @225,"LICHEN ANGREIFERN
ZU RETTEN !!";
4740 PRINT @289,"HIERMIT WIRST DU VOM VOLK ZUM ";:PRINT @321,"'MISTER WELTRAUM'
ERNANNT !!! ";
4750 PRINT @390,"WELCH EINE EHRE ...";
4760 FOR B=1 TO 20:SCREEN 0,1:FOR A=1 TO 300:NEXT A
4770 SCREEN 0,0:FOR A=1 TO 300:NEXT A,B
4780 FOR A=1 TO 3
4790 PLAY"T4O2L4.V20CL1GL16.AGFL4.GL16.AGFL4.GL16.AGFGFEFEDEDCO1L4.AL1O2E"
4800 NEXT A
4810 GOTO 4930
4820 'GAME OVER'
4830 FOR B=1 TO 10
4840 PUT(90,65)-(169,110),E,PSET
```

Dragon 32

```
4850 FOR A=1 TO 50:NEXT A
4860 PUT(90,65)-(169,110),E,PRESET
4870 FOR A=1 TO 50:NEXT A
4880 NEXT B
4890 FOR A=1 TO 5000:NEXT A
4900 COLOR 4
4910 LINE(0,0)-(255,191),PSET,BF
4920 'NEUES SPIEL ?'
4930 CLS 0
4940 RESTORE
4950 FOR A=1 TO 122:READ B:NEXT A
4960 PRINT @0,STRING$(32,131);
4970 PRINT @41,"PUNKTE :";F;
4980 FOR A=1 TO 45
4990 READ B,N
5000 SET(B,N,4)
5010 NEXT A
5020 REM DATA
5030 DATA 8,7,8,8,8,9,8,10,8,11,8,12,9,13,10,14,11,14,12,14,13,13,14,12,14,13,14
,14,14,15,14,16,14,17,14,18,14,19,13,18,12,18,11,18,10,18,10,19,9,19,8,19,7,19,6
,19
5040 DATA 6,18,5,18,4,18,3,18,2,18,2,19,2,17,2,16,2,15,2,14,2,13,2,12,3,13,4,14,
5,14,6,14,7,13
5050 FOR A=1 TO 101
5060 READ B,N
5070 SET(B,N,2)
5080 NEXT A
5090 REM DATA
5100 DATA 22,12,22,11,22,10,22,9,22,8,22,7,22,6,23,7,24,8,25,9,26,10,27,11,28,12
,28,11,28,10,28,9,28,8,28,7,28,6
5110 DATA 36,6,35,6,34,6,33,6,32,6,31,6,30,6,30,7,30,8,30,9,31,9,32,9,33,9,30,10
,30,11,30,12,31,12,32,12,33,12,34,12,35,12,36,12
5120 DATA 38,6,38,7,38,8,38,9,38,10,38,11,39,12,40,12,41,12,42,12,43,12,44,11,44
,10,44,9,44,8,44,7,44,6
5130 DATA 52,6,51,6,50,6,49,6,48,6,47,6,46,6,46,7,46,8,46,9,47,9,48,9,49,9,46,10
,46,11,46,12,47,12,48,12,49,12,50,12,51,12,52,12
5140 DATA 60,7,59,6,58,6,57,6,56,6,55,6,54,7,54,8,55,9,56,9,57,9,58,9,59,9,60,10
,60,11,59,12,58,12,57,12,56,12,55,12,54,11
5150 FOR A=1 TO 102
5160 READ B,N
5170 SET(B,N,8)
5180 NEXT A
5190 REM DATA
5200 DATA 23,20,22,19,21,19,20,19,19,19,18,19,17,20,17,21,18,22,19,22,20,22,21,2
2,22,22,23,23,23,24,22,25,21,25,20,25,19,25,18,25,17,24
5210 DATA 30,19,29,19,28,19,27,19,26,19,25,19,25,20,25,21,25,22,26,22,27,22,28,2
2,29,22,30,22,31,21,31,20,25,23,25,24,25,25
5220 DATA 33,19,34,19,35,19,36,19,37,19,35,20,35,21,35,22,35,23,35,24,35,25,34,2
5,33,25,36,25,37,25
5230 DATA 45,19,44,19,43,19,42,19,41,19,40,19,39,19,39,20,39,21,39,22,40,22,41,2
2,42,22,39,23,39,24,39,25,40,25,41,25,42,25,43,25,44,25,45,25
5240 DATA 47,19,47,20,47,21,47,22,47,23,47,24,47,25,48,25,49,25,50,25,51,25,52,2
5,53,25
5250 DATA 55,21,55,20,56,19,57,19,58,19,59,19,60,19,61,20,60,21,59,22,58,23,58,2
5
5260 PRINT @480,STRING$(31,140);:POKE 1535,140
5270 A$=INKEY$:IF A$="" THEN 5270
5280 IF A$="J" THEN 1150 ELSE IF A$="N" THEN 5290 ELSE 5270
5290 'ENDE'
5300 POKE65494,0:'NORMALE VERARBEITUNGSGESCHWINDIGKEIT'
5310 FOR A=31 TO 1 STEP -1:PLAY"T30L401V"+STR$(A)+"C":NEXT A
5320 END
5330 PLAY"T402V15L1FL4ED#L1DL4C#CO1L1BL4A#AL1G#"
5340 RETURN
```

Computronic **Kleinanzeigen** **nur gegen Vorkasse**

Bitte veröffentlichen Sie ab nächstmöglicher Ausgabe

unter der Rubrik: ☐ Angebote ☐ Kontakte ☐ Gesuche ☐ Verschiedenes

folgenden Text:

Bitte jeweils 35 Buchstaben pro Zeile! Bitte Absender nicht vergessen!

Preis für Kleinanzeigen:	Private Anzeige	DM 4,50 je Druckzeile
	gewerbl. Anzeige	DM 8,00 je Druckzeile
	Chiffregebühr	DM 10,00 je Anzeige

Name: ____________________ Straße, Nr.: ____________________

Vorname: ____________________ PLZ/Ort: ____________________

Unterschrift, Datum: ____________________ privat ☐ gewerbl. ☐
(bitte ankreuzen)

Aktion: Billige Kleinanzeigen

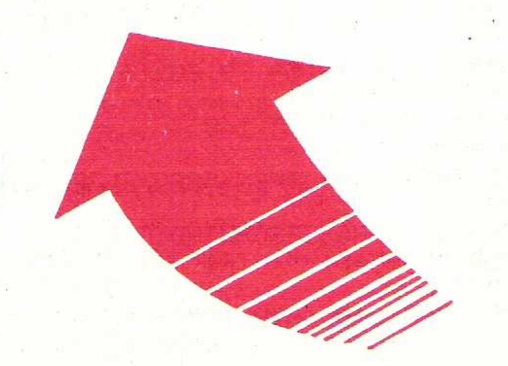
Lieber Leser,
wollten Sie eine Kleinanzeige aufgeben? Suchen Sie einen gebrauchten Home-Computer, Software oder Kontakte zu Computer-Clubs etc.? **Dann nutzen Sie jetzt den Kleinanzeigenservice von ‚Computronic'.** ‚Computronic' ist die Zeitschrift für Home-Computer-Besitzer. Sie erreichen durch uns eine Vielzahl von Lesern im In- und Ausland.

Und so wird's gemacht:
Kleinanzeigen-Coupon ausfüllen (alle Zeilen können ausgefüllt werden) und gegen **Vorkasse von DM 4.50 einsenden an:**

Tronic-Verlag
Postfach 41
3444 Wehretal 1

Eingesandt werden muß bis spätestens 11. Juli **1984** (es gilt der Poststempel).

Also, schnell ausfüllen und einwerfen in den nächsten Postkasten. Dann ist mit Sicherheit Ihre Kleinanzeige dabei.



Kleinanzeigen

TI - 99/4A
Suche EXTENDED BASIC MODUL evtl. Tausch gegen EDITORIASSEMBLER sowie Maschinensprachprogramme z.B. FLUGSIMULATOR und EX. BASIC SPIEL MINER 2049 (USA - VERSION). Tel:
Tel: 07132/37608

Suche/Tausche Software für C 64 und Atari 600/000 XL. Vorzugsweise Tausch.
Oliver Kellner,
Haydnstr. 39
6905 Schriesheim
Tel: 06223/65939

Spectrum Softw. z.B. M. Miner 8.–, ZOOM 8.–, Liste + kostl. Programm anfordern, Karte genügt:
H. Novak
Egerländer Pl. 19
6053 Obertshausen 2

C-64-Synthesizer: ca. 40 K, echte Noten, Partiturediting, Notenausdruck, Direkteinspielung+Notenumsetzung,Supersound. Info von Autor W. Kracht
Espellohweg 38
2000 Hamburg 52

ZX-SPECTRUM! Spielend BASIC lernen Ca . mit 10 Pr.+Listing+Erklärung +Tips+Tricks für den ZX 10 DM bei A. Meuser
Goethestr. 18
5205 St. Augustin
NEU-SUPER ZX Spiele-

TI 99/4A Software Service von Action bis Praxis
Info geg. 1.– Rückporto
Behringstr. 45
4600 Dortmund 50

Tausche Spectrum-Software,
Jan Weigner
Carl-Schurz-Straße 11
2800 Bremen
Sinclair forever!!!

TI-99/4A : ca. 90 TI-Basic und ca. 80 Ex - Basic Programme zu verk. Info g. Rückumschlag
B. Knedel
Tulpengasse 16
3171 Weyhausen
Tel. 05362/71187

An alle SF- und Fantasy - Freunde ! Jeweils 8 Mitspieler für Strategiespiele gesucht. Mit Gewinnchancen. Näheres gegen Freiumschlag von :
H. Topf
A. Bucherer Str 63
5300 Bonn 1

Roulettesoftware f. Sharp PC1211/12/4551/PC 1500
Roulettesystemprogramme
»Auf Dauer gew. «
»AD Astra« »Number one«
»Acclaim« »Plein Prive«
»Accord« DM 550.- Cass
B. Nowotny
Giselastr. 3
8000 München 40

Suche Seikosha 6P-100A Drucker mit Interface für ZX Spectrum
Lothar Schröder
Dellchenstrasse 3
6731 Frankeneck

VZ 200 Softw. Info g. Freiumschlag
W. Fischer
Friedrichstr. 14
6463 Freigericht 5

Orig. engl. Software von EMJAY, (Wargame News, The War Machine), für Commodore - Rechner ab 16k und CBM 64 ! Näheres gegen Freiumschlag von :
H. Topf
A. Buchererstr. 63
5300 Bonn 1

Verkaufe
Anrufbeantworter,
Drahtlostelefon,
Telefon - Anlage
Eurosignal 1200.-DM
Div. USA-Telefone
Info bei:
Reinhard Winner
Höchbergerstr. 62
87 Würzburg

100 Programme für VC-64 wegen Hobby - Aufgabe zu verkaufen !
Rolf Freitag
Gneisenaustr. 87
4600 Dortmund 1
Tel: 0231 - 825826

COMPUTER-CASSETTEN
im 10er Pack, BASF-Band-LHD, mit Box, Etikett und Einleger
C 10 nur 15,– DM
C 20 16,– DM; C 30 17,– DM
Brandneu von TDK:
PC-10 im 10er Pack 29,– DM
CASSETTEN-AUFKLEBER
auf Lochstreifen 100 St. 5,– DM
120 St. auf A4-Druckbögen
7,– DM; Versand ab 20,– DM:
Christomenia-Cassettenstudio
Postfach 20
3584 Zwesten, Tel. 0 56 26-2 81

100 neue Programme
für MZ 80 K, MZ 80 A, MZ 700, VZ 200, Laser 210, TI 99.
Info, 1,– DM in Marken, von H. Weisel
Königsberger Str. 20
5412 Ransbach

Commodore 64
irre Programme (375) zu irren Preisen !
Ausführliche Liste (48 Seiten) gegen DM 3,- in Briefmarken.
Detlef Treichel
Mittelbruchzeile 105
1000 Berlin 51

ZX Spectrum
Soft- und Hardware.
Preiswert durch Direktimport.
Info bei:
Ursula Kunz
Junge Hälden 3
7500 Karlsruhe 41

Kontakte:
Comp.-Club Bad Hersfeld sucht Mitglieder !
Kontakt über
Roland Reyer
Am Giegenberg 21
6438 Ludwigsau 1
oder Jugendhaus Hersfeld

A C H T U N G
für TI 99/4A jetzt lieferbar

EXTENDED BASIC	DM 295,-
Editor Assembler	DM 189,-
Parsec	DM 79,-
TI-Logo II	DM 320,-
32K-Erw. extern	DM 428,-

Modul-Spiele direkt aus den
* * * * USA * * * * * * *
zb. Mash, Demon, Baseball, Tunnels of Doom, Moon mine, Jawbreaker, Hopper, Attack, Bigfoot ...... DM 99,-
diverse ATARI-Spiele für TI Pac Man, Donkey Kong, Dig Dug, Mrs. Pac Man, Defender, Centipede ... DM 99,-
Neu!!! MBX-Sprachsysteme nur für den TI-99/4A
Spiele direkt mit der Sprache steuern DM 298,-
Neu !!! GRAFIK TABLEAU

für TI 99/4A	DM 298,-
für ATARI	DM 248,-
für VC 64	DM 298,-

Bitte Preisliste anfordern:
RADIX Bürotechnik
2000 Hamburg 13
Bornstr. 4
Tel. 040/44 16 95 10-18.00 h

Service

Jedes Programm in Computronic wird einer Endkontrolle in unserem Hause unterzogen und genauestens geprüft. Alle im Heft abgedruckten Programme sind nach der richtigen Eingabe der Listings auch funktionsfähig. Viele Leser verlieren jedoch schnell die Geduld am Programmieren, sollte etwas einmal nicht klappen. Die häufigste Ursache von Störungen im Programm, sind unterlaufene Fehler bei der Eingabe. Verzweifeln Sie nicht, sollten Sie einmal keine Zeit zum Programmieren haben oder sollte etwas nicht gelingen. Alle Programme werden im Verlag gespeichert und können jederzeit mit beiliegender Bestellkarte zum Selbstkostenpreis bezogen werden.

Tragen Sie bitte alle notwendigen Angaben in die Bestellkarte ein, und senden sie an:

Tronic-Verlag GmbH, Postfach 41, 3444 Wehretal 1

Preisliste Heft 4 (K) (D)

VC-64 K = 8,– DM D = 15,– DM
Mauern, Widerstand

ZX-Spectrum
Inventur K = 12,– DM

TI 99 K = 8,50 DM
Drei-Kronen-Spiel
Zahlenputzen

VC-20 K = 11,50 DM D = 18,– DM
Hangman, Saurer Regen, Quadr. Gleichung

ZX-81 K = 8,50 DM
Invader, Gun-Man
Landeanflug Boing K = 15,– DM

Dragon 32 K = 6,– DM
Blizzard

Apple II K = 14,50 DM D = 19,50 DM
Wilder Westen,
Karambolage,
Maskengenerator

Atari K = 10,50 DM
Mastermind,
Schlangenkrieg

Preisliste Heft 5 (K) (D)

TI 99 K = 14,50 DM
Karl der Käfer
Alien-Landing

VC-64 K = 15,50 DM D = 23,50 DM
Space-Comets
Erdspalte
Sprite-Data

Apple II K = 9,50 DM D = 17,50 DM
Musik-Maker
Mission-Adler
Disk-Catalog

Sharp MZ 700 Kalender K = 8,50 DM
Sharp PC 1500 Lotto

Dragon 32 K = 10,– DM
Space-Flight, Geosoft

ZX-81 K = 10,– DM
Go-Ball, Grand-Prix

ZX Spectrum K = 6,50 DM
Missile-Comment

Atari K = 12,50 DM
Tank-Battle
Oil Panic

Preisliste Heft (K) (D)

VC-64 K = 16,50 D = 23,50
„Data Generator"
„Autostart"
„Bestellschein"
„Roadpainter"

Dragon 32/64 K = 8,50
„Wargames"

Apple II D = 19,50 K = 12,50
„Snake"
„Super Datei"
„Shape-tables"

VC-20 K = 8,50 D = 15,00
„Bestellschein"
„Glücky"

ZX-81 K = 10,00
„Moon-Crash"
„ZX-Draw"

ZX-Spectrum K = 13,50
„Defender"
„Lui der Wurm"
„Alternativer Zeichensatz"

TI-99 K = 14,50
„Jack the Digger"
„Noah - 2099"

Atari K = 12,50
„Tomstone-City"
„Schneevogel"

Preisliste Heft 7 (K) (D)

VC-64 K=15,50 DM D=23,50 DM
Hardcopy
Space-Fighter
Data-Generator

ZX-81 K=10.- DM
Tonprogramm
Aldebaran

Atari K=12,50 DM
Startup
Zeilen-Split
Chopper-Fight

VC-20 K=11.- DM D=15,50 DM
Multigraph
All-Rammer

ZX-Spectrum
Matheprogramm
Bongo-Beecatcher K=12,50 DM

Apple 2
Library
Fight K=12,50 DM

Dragon 32
Laser-Attack K=7,50 DM

Computronic Bestellkarte-Cassettenservice

Alle im Heft abgedruckten Programme können als zusätzlicher Service, zum Selbstkostenpreis, über den Verlag bezogen werden.

Die Zustellung erfolgt: gegen **Vorkasse** ☐ / per **Nachnahme** + Versandkosten ☐ innerhalb von 14 Tagen!

Achtung: Disketten können vorerst nur für: Apple
VC-64
VC-20 bezogen werden!

Entnehmen Sie bitte aus unserer Preisliste Seite 66 die notwendigen Angaben für Ihre Bestellung:

Bitte liefern Sie mir: ☐ Cassette für ________ (System) ☐ Anzahl

☐ Diskette für ________ (System) ☐ Anzahl

zum Preis von gesamt: ________ DM

Name: ________ Straße, Nr.: ________

PLZ/Ort: ________ Unterschrift, Datum: ________

Computronic Abonnement

Bitte ausschneiden und senden an: Tronic-Verlag, Postfach 41, 3444 Wehretal 1.

Hiermit bestelle ich die Zeitschrift »Computronic« ab Heft Nr. ______ zum Jahresabonnementpreis (12 Ausgaben) von DM 55,— incl. Versand, Inland und DM 75,— incl. Versand, Ausland.

Name/Vorname: ________

Straße, Nr. ________ Plz, Ort: ________

Ich wünsche folgende Zahlungsweise:

☐ Bargeldlos durch Bankeinzug: Bankleitzahl: ________

Geldinstitut: ________ Konto Nr.: ________

☐ gegen Rechnung

☐ gegen Vorkasse
(betreffendendes bitte ankreuzen)

____
Datum, Unterschrift

Abonnements-Kündigungen: 6 Wochen vor Ablauf des Jahres-Abonnement.

Korrekturen

Atari »Tomstone-City

Richtig

```
100 DIM A$(44),KX(150),KY(150),S(15),FX(
6),FY(6):CH=(PEEK(106)-8)*256
110 IF PEEK(CH+8)<>24 THEN POKE 559,0:GO
SUB 32000
1000 GRAPHICS 17:POKE 710,148:POKE 709,0
:POKE 708,54:POKE 711,180:POKE 712,238:P
OKE 756,CH/256
31000 FOR A=10 TO 14 STEP 2:POSITION 8,A
:? #6;CHR$(168);" ";CHR$(168);" ";CHR$(1
68):NEXT A:RETURN
32010 SOUND 0,0,0,0:X=USR(ADR(A$),57344,
CH,1024)
32030 FOR A=0 TO 7:READ B:SOUND 0,B,10,6
:POKE CH+C*8+A,B:NEXT A:GOTO 32020
```

VC-64 »Roadpainter«

Richtig

```
62335 DATA255,255,255,7,7,135,199,231

63005 DATA142,2,208,234,234,234,173,250,7,201,13,240,20,201,15,240,28,201,14
63006 DATA240,36,174,5,208,32,62,128,142,5,208,76,194,128,174,4,208,32,53,128
63007 DATA142,4,208,76,194,128,174,5,208,32,53,128,142,5,208,76,194,128,174

63021 DATA162,5,189,96,5,201,57,240,4,254,96,5,96,169,48,157,96,5,202,76,2
63022 DATA144,162,3,189,40,6,201,48,208,9,224,0,240,4,202,76,24,144,96,162
63023 DATA3,189,40,6,201,48,240,4,222,40,6,96,169,57,157,40,6,202,76,42,144
63024 DATA32,0,144,76,74,144,32,0,144,32,22,144,162,0,169,0,157,0,212,224,24
63025 DATA240,4,232,76,78,144,169,11,141,24,212,141,0,212,141,1,212,169,4
```

Zu dem Programm SPLIT:

Folgende Zeilen müssen geändert werden (▬=Blende)
500 IF A$(J,J+3)="REM▬" OR A$(J,J+2)="IF▬"
OR A$(J,J+4)="DATA▬" THEN J=LEN(A$):GOTO 540
860 IF NOT (A$(J,J+3)="▬ON▬" OR A$(J,J+3)=":ON▬")
THEN 930
940 IF A$(J,J+3)="REM▬" OR A$(J,J+4)="DATA▬"
THEN 1060

Diese Änderungen sind nötig weil
–In Zeile 860 versucht wurde aus POSITION ein ON...GOTO zu machen.
–in den Zeilen 500 u. 940 in einer Data-Zeile nach Befehlen gesucht wurde.
Diese Fehler sind mit dem angegebenen Änderungen nicht mehr möglich.

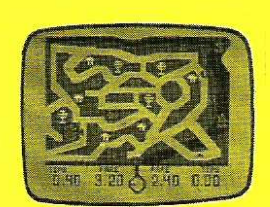
Heft 8 von Computronic ist:

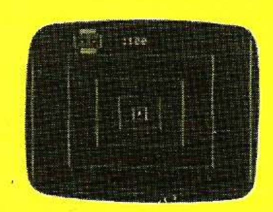
Ab 30. Juli bei Ihrem Zeitschriftenhändler erhältlich!

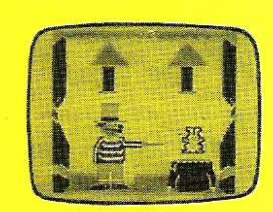


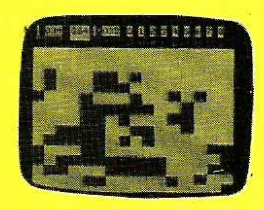
Natürlich bringen wir dann wieder
jede Menge Programme!
Programme!
Programme!





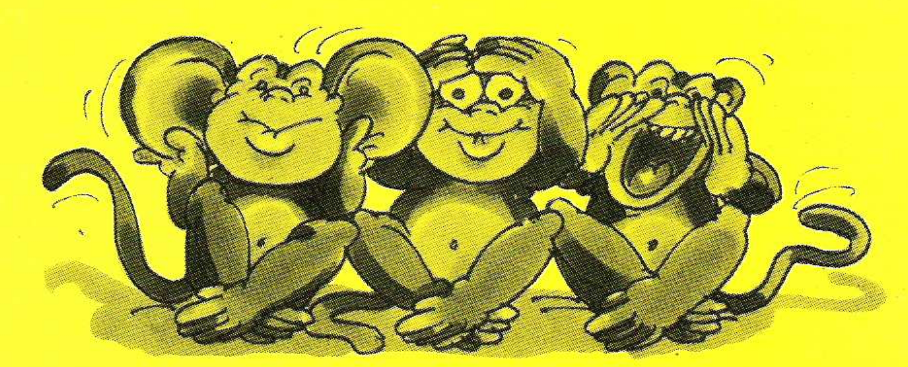






und:

Neues, Infos, Tips + Tricks!



Es verabschiedet sich: Die Computronic-Redaktion

Impressum: **Computronic** erscheint monatlich im Verlag: Tronic-Verlagsgesellschaft mbH, Landstraße 29, 3444 Wehretal 1, Tel.: (05651) 40643/40693. **Redaktion:** Rainer Beck (verantwortlich), H. A. Ederg. **Freie Mitarbeiter:** Frank Brall, G. Wagner. **Satz und Reproduktion:** Intragrafik . . **Druck:** Schimmel KG, Würzburg. **Vertrieb:** Inland (Groß-, Einzel- und Bahnhofsbuchhandel), sowie Österreich und Schweiz: Verlagsunion, 6200 Wiesbaden. **Anzeigenverwaltung:** E. Herwig. **Anzeigenpreise:** Es gilt die Anzeigenliste Nr. 1. Bitte Media-Unterlagen anfordern. **Bezugspreise:** Einzelheft (Inland) DM 5,50, Abonnement (Inland) DM 55,—, (Ausland) DM 75,—. **Autoren und Manuskripte:** Bei Zusendung von Manuskripten und Datenträger erteilt der Autor dem Verlag die Genehmigung zum Abdruck und Versand der veröffentlichten Programme auf Datenträger. Für die mit Namen des Verfassers gekennzeichneten Beiträge übernimmt die Redaktion lediglich die presserechtliche Verantwortung.



computronic
Infos

Tips und Tricks für Einsteiger, Anfänger und Fortgeschrittene. Informationen und Daten. Sinnvolle Anwendungen und Anregungen. Lernen, anhand von Computronic, Programme einzugeben und Programme zu schreiben. Von Basic bis Maschinensprache. Neues vom Computermarkt über Hard- und Software.

computronic
Software-Listings

Unser Schwerpunkt in den Ausgaben (70 %). Leichte und anspruchsvolle Programme werden im Heft abgedruckt, wobei die Redaktion alle bekannten Home-Computer berücksichtigen wird. Wenn man will, kann man die Programme einfach eingeben, damit spielen oder arbeiten. Alle Konzepte und Begriffe sowie alle Spiele werden vor ihrer Anwendung erklärt.

computronic
Software-Service

Abgedruckte Programme können als zusätzlicher Service über den Verlag bezogen werden. Wer kein Interesse hat seinen Computer zu »füttern«, oder wem es zu langweilig wird zu programmieren, kann alle abgedruckten interessanten Spiele auf Cassette und Diskette erwerben.

Computronic erscheint monatlich im Tronic-Verlag. Der Bezugspreis beträgt im Inland DM 5,50. Im Preis ist die gesetzliche Mehrwertsteuer enthalten.

Tronic-Verlag GmbH · Postfach 41 · 3444 Wehretal 1 · 05651-40643/40693